CERVICAL CANCER CLASSIFICATION USING DEEP LEARNING TECHNIQUES

BY

ZAID MOHAMMED ATEF AL-YAFEAI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

DECEMBER 2018

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ZAID MOHAMMED ATEF AL-YAFEAI** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

Dr. Khalid Al-Jasser

Department Chairman

Dr. Lahouari Ghouti (Adviser)

Dr. Wasfi G. Al-Khatib (Member)

Dr. Salam A. Zummo

Dean of Graduate Studies

Dr. Nabil M. Maalej (Member)

20/12/2018

Date

عمادة الدراسات العليا

DEANSHIP OF GRADUATE STUDIES

Dedication

To my mother, father and sisters.

ACKNOWLEDGMENTS

I want to thank my advisor for his continuous support, patience, motivation and immense knowledge. I also extend my special thanks to the committee members for their valuable time and support. Special thanks to the department of computer science, dean, chairman and faculty. A special thank for my brother Akram Al-Absi for his help and gaudiness.

TABLE OF CONTENTS

ACKNOWLEDGMENTS v

LIST OF TABLES ix

LIST OF FIGURES x

ABSTRACT (ENGLISH) xiii

ABSTRACT (ARABIC) xv

CHAPTER 1 INTRODUCTION 1

1.1 Introduction 1

1.2 Problem Statement 3

1.3 Thesis Contributions 5

1.4 Thesis Organization 6

CHAPTER 2 LITERATURE REVIEW 7

2.1 Cervical Cancer 7

2.2 Cervical Cancer: Causes, Detection and Cures 8

2.3 Computer-Aided Detection and Diagnosis of Cervical Cancer 8

2.3.1 Non-machine Learning Approaches for Cervical Cancer Segmentation and Classification 8

2.3.2 Cervical Cancer Segmentation and Classification 9

2.3.3 Segmentation and Region of Interest :General approaches 11

CHAPTER 3 DEEP LEARNING APPROACHES 13

3.1 Introduction: Automatic Feature Extraction 13

3.2 Convolutional Neural Networks (CNNs) 14

3.2.1 Convolutional layers 15

3.2.2 Pooling layers 16

3.2.3 Non-linearities 16

3.2.4 Weight Regularization 18

3.2.5 Dropout 19

3.2.6 Classification Layers 19

3.3 From CNNs to Deep Neural Networks (DNNs 21

3.3.1 Deep Networks 21

3.3.2 Batch-Normalization layers 22

3.3.3 Backpropagation and Update rule 23

3.4 Common Deep Learning Frameworks for Classification 24

3.4.1 Google Inception Models 24

3.5 Common Deep Learning Segmentation and Detection Models 27

3.5.1 Evaluation Metrics 27

3.5.2 Object Detection Models 27

3.5.3 Object Segmentation Models 30

CHAPTER 4 FULLY-AUTOMATED DEEP LEARNING PIPELINE 37

4.1 Introduction 37

4.2 Proposed Segmentation Approach 39

4.2.1 Data Preparation 39

4.2.2 Performance Measure 40

4.3 Proposed Classification Approach 42

4.3.1 Preprocessing 42

4.3.2 Performance Measure 42

4.3.3 Training and Validation 44

4.3.4 Hand-crafted Features 45

4.3.5 CNN Features 49
4.3.6 Classifier 50

CHAPTER 5 EVALUATION OF THE PROPOSED APPROACH 53

5.1 System Configuration 53
5.2 Dataset 54
5.2.1 Overview 54
5.2.2 Data Collection 58
5.2.3 Data Distribution 60
5.2.4 t-SNE Clustering 62
5.2.5 Data Augmentation 68
5.3 Results and Discussion 70
5.3.1 Segmentation 70
5.3.2 Hand Crafted Features 75
5.3.3 Automatic Feature Extraction 77
5.3.4 Training 77
5.3.5 Failed Cases 80

CHAPTER 6 CONCLUSION AND FUTURE WORK 84

6.1 Conclusion 84
6.2 Future Work 85

APPENDIX A MULTI-CLASSIFICATION 86

APPENDIX B IMBALANCED CLASSIFICATION 90

REFERENCES 92

VITAE 101

LIST OF TABLES

2.1 Baseline results from different Segmentation benchmarks in the literature. 11

2.2 Baseline results from different classification benchmarks in the literature. 12

4.1 Yolo parameters 41

4.2 True positive, true negative, false positive and false negative. 43

5.1 Description of the fields in the dataset. 55

5.2 Worst Histology. 58

5.3 Extracted balanced dataset. 61

5.4 Comparing IoU and prediction time against other methods in the literature. 71

5.5 Model 1: hand crafted features with 1 hidden layer, Model 2: hand crafted features with 2 hidden layers, Model 3: CNN with 2 conv layers and 1 hidden layer, Model 4: CNN with 3 conv layers and 2 hidden layers 78

LIST OF FIGURES

1.1 Cervigrams for each grade of cervical cancer. 3

1.2 On the left we have some negative samples and on the right column we have some positive samples. 4

3.1 A neural network. 14

3.2 LeNet architecture by LeCun [1]. 15

3.3 Dropout [2]. 19

3.4 Inception v3 basic modules. 24

3.5 Inception v3 complete model [3]. 25

3.6 Inception v4 basic modules. 25

3.7 On the left is the overall structure and on the right is the inner layers of the stem [4]. 26

3.8 R-CNN architecture. 28

3.9 Fast R-CNN architecture. 29

3.10 Faster R-CNN architecture. 30

3.11 An example of segmenting a cervix. 31

3.12 The architecture of a U-Net model [5]. 32

3.13 The architecture of a SegNet model [6]. 33

3.14 The architecture of a Mask R-CNN model [7]. 34

3.15 YOLO architecture [8]. 36

4.1 The pipeline of our architecture. 38

4.2 Visualization of the bounding box parameters. 40

4.3 Visualization the ROC graph and AUC values for random, good and excellent classifiers. 44
4.4 L*a*b* color space. 45
4.5 An example of a histogram for the the set of values (2.3, 3, 2.5, 0, 0.5, 1.5, 2, 0.3, 2.9). 46
4.6 PLAB descriptor. 47
4.7 LBP procedure. 48
4.8 PLBP descriptor. 48
4.9 PHOG descriptor. 49
4.10 CNN features extracted from a model with two conv layers. 51
4.11 CNN features extracted from a model with three conv layers. 52
4.12 A binary classifier. 52

5.1 Comparisons of the fields in Table 5.1 with respect to the worst histology analysis. Each color represents different histology value. Each graph is a comparison between two fields in the data as illustrated in the x and y coordinates. The main diagonal counts the number of occurances of each data field. 56
5.2 Comparing HPV and worst histology. The main diagonal represents the number of occurrences of each data field for each respective value. The other graphs compare between the histology value and the HPV status. 57
5.3 On the left we have some negative samples and on the right column we have some positive samples. 59
5.4 Distribution of the labels for the histology field. 60
5.5 Distribution of the labels for the histology field with a simple image per patient. Since each patient has multiple images we only extract one image for each patient to avoid data leaks in the validation set. . . 61
5.6 t-SNE distirubtion for the first color channel. The x coordinate represents the first component in the projected space and the y axis is the second component in the projected space. 63

5.7 t-SNE distribution for the second color channel. 64
5.8 t-SNE distribution for the third color channel. 64
5.9 t-SNE distribution for all RGB color channels. 65
5.10 t-SNE for PLBP features. 65
5.11 t-SNE for PHOG features. 66
5.12 t-SNE for PLAB features. 66
5.13 t-SNE for PLAB+PHOG+PLBP features. 67
5.14 256 x 256 random cropping followed by horizontal flipping followed by 45 degree rotation. Since we are applying random cropping some features might not exist in the current augmented batch but will exist in other batches in different epochs. 69
5.15 The confidence scores of some cervigrams bounding boxes using YOLO. 73
5.16 More confidence scores samples. 74
5.17 ROC graph with AUC values for each model. 79
5.18 Confidence values for all the data. 80
5.19 Some bounding box predictions with less than 80 % confidence. 81
5.20 Samples of false positive and false negative cervigrams with the probability value. 83

A.1 Distribution of the labels for the histology extracted from our dataset. 86
A.2 Comparisons between the four models with 3 classes. 88
A.3 Distribution of the labels for 5-way classification. 89
A.4 Comparisons between the four models with 5 classes. 89

B.1 Distribution of the labels for the histology field for the full dataset. . . 90
B.2 Comparisons between the four models with two classes. 91
B.3 Comparisons between the four models with five classes. 91

THESIS ABSTRACT

NAME: Zaid Mohammed Atef Al-Yafeai

TITLE OF STUDY: Cervical Cancer Classification using Deep Learning Techniques

MAJOR FIELD: Computer Science

DATE OF DEGREE: December/2018

Cancer is among the top causes of deaths among women worldwide. These fatalities are mainly attributed to breast and cervical cancers. Breast cancer has attracted a lot of attention in the medical and engineering fields due to its widespread public interest. Advances in the early detection of the cancer tumors have boosted the survival rates after cancer occurrence. Despite these advances, cervical cancer has not received attention similar to that given to breast cancer. The increase in incidence and mortality rates corroborate this fact. More importantly, these rates are higher in less developed regions in the world. Cost-effective and automated approaches for the detection and classification of cervical cancer will certainly contribute to a drastic decrease in mortality rates. Machine learning, and more specifically deep learning, approaches gained maturity in solving challenging medical imaging problems. State-of-the-art im-

age segmentation and classification paradigms using deep learning make the design and implementation of fully-automated segmentation and classification pipelines for cervical cancer a reality. This thesis contributes in this direction by proposing a novel fully automated system for the segmentation of cervical images, known as cervigrams, and classification of cervical cancer tumors. Various deep learning models are proposed and evaluated in terms of: 1) accurate segmentation of the region-of-interest (RoI) near the tumor area 2) correct classification of the cervical tumor with 77% *accuracy. In addition, the automation improvement is measured using the overall system speed. Our proposed models are faster by a factor of* 10^3 *compared to manual and semi-automated approaches. The overall pipeline efficiency comes at the cost of a negligible decrease in tumor classification accuracy.*

ملخص الرسالة

الاسم: زيد محمد اليافعي

عنوان الدراسة: تصنيف وتقسيم سرطان عنق الرحم بإستخدمام تقنيات التعليم المتعمق

التخصص: علوم الحاسوب

تاريخ الدرجة العلمية: ربيع الآخر / 1440

يعتبر السرطان من أهم مسببات الوفيات في النساء حول العالم. هذه الوفيات تكون مرتبطة غالباً بسرطان الثدي و سرطان الرحم. سرطان الثدي كسب الكثير من الإهتمام من الجانب الطبي والهندسي لإنتشاره. التطور في مجال اكتشاف السرطان إدى لزيادة حالات النجاة. بالرغم من هذه التطورات لم يكسب سرطان الرحم ما يستحقه من الإهتمام. زيادة حالات الوفيات يدعم هذه الحقيقة. بشكل أكثر أهمية، حالات الوفية تزداد في الدول الغير متطورة. الطرق الغير مكلفة الآلية في اكتشاف سرطان الرحم سيقوم بتطوير حياة الناس والتقليل من حالات الوفيات بشكل عام. تعلم الآلة والتعليم المتعمق تطورت بشكل كثير في مجال حل المشاكل الطبية المعقدة. هذه الرسالة تساهم في هذا المجال من خلال تطوير طريقة جديدة في مجال تصنيف واكتشاف سرطان الرحم. نقدم العديد من النماذج المقترحة لأكتشاف منطقة السرطان وكذلك تصنيفها بشكل صحيح. الطريقة المقترحة تعتبر أسرع ب1000 مرة عن الطرق المتوفرة حالياً. هذا بشكل غير مباشر يؤثر على جودة النظام

في تصنيف السرطان ولكن هذا الهبوط في الكفاءة ليس له هذا التأثير الكبير في النظام العام.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Cancer arises in the body when the cells of a specific organ start to grow abnormally. Cancerous cells can affect many organs like the brain, the lungs, etc. Cervical cancer is the cancer that attacks the cervix of a woman. The cervix is the neck-shape passage at the bottom of the uterus. In developing countries, cervical cancer is ranked third as the most fatal type of cancer [9]. In 2012 around half a million incidences of cervical cancer were diagnosed and nearly half of that number of deaths is estimated [9]. Around 700 deaths happen each day due to cervical cancer [10]. These numbers seem to be only rising as it is expected that the number of deaths to reach around 400,000 by 2030 [11]. Cervical cancer screening is the process by which a test is performed to check the existence of abnormal tissues or cancerous cells in the cervix. Screening can help curing cervical cancer by the detection of cervical intraepithelial neoplasia (CIN) which indicates abnormal changes in the cervix. CIN can be separated into the following categories: CIN1, CIN2 and CIN3. While, CIN1 needs observation

only, CIN 2/3+ require treatment. Physicians rely on different screening methods to differentiate between these types to decide if the patient needs treatment or not. Current screening methods include Pap tests, human Papillomavirus (HPV) testing and visual inspection [12]. A Pap test defines the process of taking a sample from the cervix and inspecting it under a microscope. However, the Pap test suffers from 6-55 % false-negative rate [13]. The HPV test is a DNA test that detects cervical cancer by associating it with a specific human Papillomavirus . Usually this test is not recommended as it suffers from a high false positive rate [14]. Furthermore, the cost of such tests is quite high. In developing countries, it is difficult to afford these tests hence they depend on visual inspection. However, visual inspection can be tricky and requires expertise which lacks in such countries. Cervix shape, color and texture can help physicians decide which treatment to be taken thereafter [15]. Hence, detecting these types is related to the expertise of the physician which is not available in developing countries. Digital Cervicography refers to the process of capturing a photo of the cervix named (cervigram). Usually this is done after the application of 5% of acetic acid. Hence, automated segmentation and classification can be used on cervigrams.

1.2 Problem Statement

We depend on visual inspection in order to detect cervical cancer. Given a cervigram image we need to classify it into one of two categories. The first one is either normal or mild intraepithelial neoplasia called (CIN1). We consider this as the negative class. CIN1 does not require further treatment. The second category is moderate/sever intraepithelial neoplasia (CIN 2/3+). This also includes cancer which is usually considered to be CIN4. Figure 1.1 shows a cervigram for each grade of cancer.

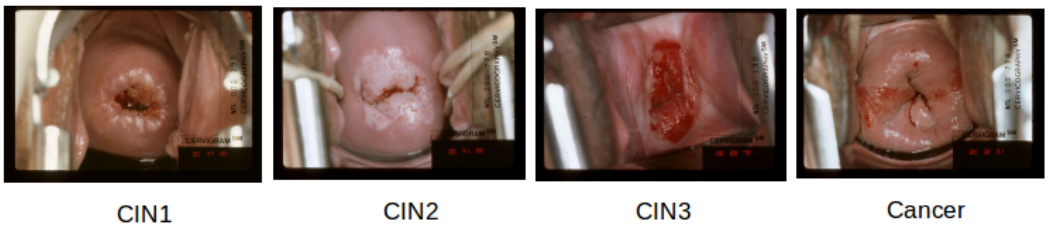

Figure 1.1: Cervigrams for each grade of cervical cancer.

It is also important to extract the region of interest from the images. As we see from the images in Figure 1.2 that there are some features in the images that might cause our classifiers to fail. For instance, the images include a black boundary that increases the size of the images and isn't a feature of a certain class. Moreover, there are some medical equipment that appear in the some images which might cause our classifiers to get confused. So, one important task in our work is to able to extract the region of interest which mostly contribute to the classification of the process.

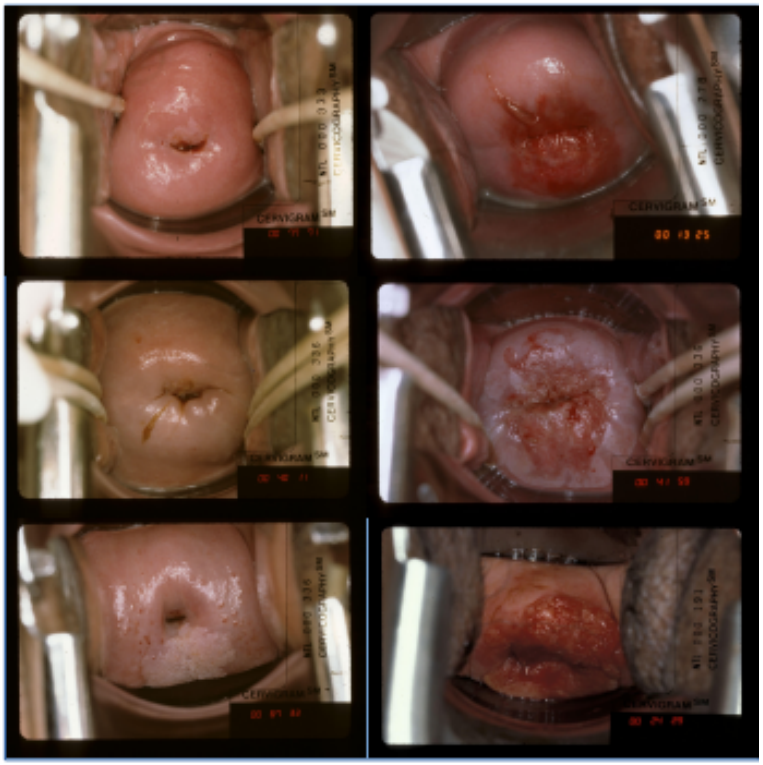

Figure 1.2: On the left we have some negative samples and on the right column we have some positive samples.

1.3 Thesis Contributions

As illustrated in the previous section we want to classify each image into one of two categories: positive class or negative class. We provide an overall architecture to solve cervical cancer segmentation and classification. We suggest a fully automated approach for cervical cancer segmentation and classification using neural networks. The process contains data collection, segmentation, preprocessing, augmentation, feature extraction and classification. First, we collect a labeled dataset that associates each cervigram with either positive or negative class. We then provide a segmentation approach for detecting the area of interest in the cervigram. Our approach is faster compared to other procedures in the literature with a comparable performance. Furthermore, our approach's segmentation time depend on the size of the dataset we are training on. We prove that our segmentation approach is 10^3 better than the methods available in the literature for such domain. After that, we compare between different classifiers for cervix classification. The first approach is using hand-crafted features using three descriptors. The hand-crafted features are a pyramid of locally binary patterns (PLBP), a pyramid histogram in the L*a*b* color space (PLAB) and a pyramid histogram of oriented gradients (PHOG). We compare this approach to automated approaches using convolutional neural networks (CNNs) feature extractions. We use various metrics to compare the two approaches and analyze the results.

1.4 Thesis Organization

We start by a literature review of cervical cancer in chapter 2. We discuss the known causes, non-computer aided detection methods and the cures available in the literature. We summarize the approaches in the literature that use computer-aided approaches for the diagnosis of cervical cancer. We discuss different machine learning and non-machine learning approaches. The next chapter explains deep learning approaches in general for computer vision classification and segmentation. We discuss the general architecture of convolutional neural networks (CNNs) including the different layers, activation functions, regularization approaches and finalize with classification layers. We then describe in details deep learning architectures for classification and segmentation like AlexNet, VGG, Inception, ResNet, YOLO, etc ... In chapter 4 we describe a fully automated approach for cervix segmentation and classification. We discuss, the proposed approach, training and hyper-paramters tuning and validation and testing. Chapter 5 discusses the evaluation approach for the discussed pipeline. We discuss the experiment setup, system configuration and the used dataset with appropriate analysis. We also evaluate our approach for segmentation and classification. We compare our segmentation approach to other methods in the literature. We also compare hand-crafted features with convolutional neural networks for classification.

Finally, we conclude the thesis with recommendations and possible future work to build on our work.

CHAPTER 2

LITERATURE REVIEW

2.1 Cervical Cancer

The cervix is the neck-shape passage at the bottom of the uterus. In developing countries, cervical cancer is ranked third as the most fatal type of cancer [9]. Also, Cervical cancer is considered the second most common type of cancer in women with age range in (15, 44) years around the world [16]. More than 80% of deaths attributed to the disease occur in developing countries [16]. In 2012 around half a million incidences of cervical cancer were diagnosed and nearly half of that number of deaths is estimated [9]. Around 700 deaths happen each day due to cervical cancer [10]. These numbers seem to be only rising as it is expected that the number of deaths to reach around 400,000 by 2030 [11].

Cervical cancer could be divided into three different types. The mild intraepithelial neoplasia (CIN1) does not need treatment. While moderate/sever intraepithelial neoplasia (CIN 2/3+) require treatment.

2.2 Cervical Cancer: Causes, Detection and Cures

Epidemiological studies have proven that the relation of human papillomavirus (HPV) to cervical cancer is evident. These results are separated from other risks and interestingly consistent in several countries around the world [17]. Moreover, Walboomers et al. show that is a major reason for invasive cervical cancer around the globe [18]. Different samples taken from different geographic regions show prevalence in cancer biopsy associated with papillomavirus DNA from a cervical carcinoma [19]. Radiotherapy and chemotherapy that contain cisplatin increase the rates of survival among women with locally advanced cervical cancer [20].

2.3 Computer-Aided Detection and Diagnosis of Cervical Cancer

In this section we discuss the computer aided approaches for the detection and diagnosis of cervical cancer.

2.3.1 Non-machine Learning Approaches for Cervical Cancer Segmentation and Classification

Mange discusses the use of computer-based algorithms to detect cancerous cell in the cervix [21]. It suggests the usage of PAPNET cytological screening system for the detection of abnormal cells. It uses the conventional PAP smears along with a neural network for the automation of the process of detecting precancerous tests.

Other methods suggest using segmentation to extract the cervical cell nuclei images. Bamford and Lovell suggest the use of an active contour method for the extraction of the cervical cell nuclei [**?**], [**?**]. A region of interest is first identified then a specific number of contours are extracted. After that, a specific algorithm is used to extract the most relevant contours [22]. On the other hand, other methods suggest the use of feature extraction techniques. Chang et al. suggest using the size and deformation of the cell nuclei to categorize the cell nuclei as abnormal [23]. A pre-processing method is first applied to get rid of the noisy parts of the image then the cell nuclei is extracted [23]. They suggest using two complementary approaches to classify the cells which are gray-level and energy method to extract the abnormal cells [23]. Kim and Huang suggest using an optimized bounding box method to segment the cervix. K bounding boxes are extracted from similar images using a similarity metric and the best one is chosen using a combination of Euclidean distance and intersection over union metrics [24]. Song et al. used a Sobel filter to detect the cervix and a multi-model approach for classification [25]. The classification methods collects information from cervigrams and different clinical tests to reach a conclusion about the patient. They evaluate a similarity measure on the data level and cervigrams level to extract a label for the current patient.

2.3.2 Cervical Cancer Segmentation and Classification

Song et al. compared the quality of two approaches for classification: majority voting method and support vector machines (SVM) [25]. These classifiers operate on hand crafted features that include colour features and texture features . Kim and

Huang compared between automated feature extraction classifiers like support vector machines(SVM), random forests, convolutional neural networks (CNN), etc ... and hand-crafted features like pyramid of locally binary patterns (PLBP), pyramid histogram in the L*a*b* color space (PLAB) and pyramid histogram of oriented gradients (PHOG) [24]. The segmented image is first re sized to (300, 250) as suggested by [24]. Color and texture of the cervix seem to be good descriptors for cervix classification as suggested by [24]. PLAB, calculates a pyramid of histograms in the L*a*b* color space space. The image is first converted to the L*a*b* space. Then a histogram is evaluated at different pyramids by divindg the image into different levels. The second descriptor is a pyramid of local binary pattern (PLBP). At each pixel located at (x_c, y_c) calculate the local binary pattern

$$LBP(x_c, y_c) = \sum_{p=0}^{7} s(i_p - i_c)2^p \tag{2.1}$$

Where i_c is the middle pixel value and i_p corresponds to the gray scale value of the neighbor pixel. Note that each center pixel where have 8 neighbors $c \in \{0, 1, \cdots, 7\}$. $s(x)$ is a sign function defined as

$$s(x) = \begin{cases} 1 & x \geq 0 \\ \\ 0 & x < 0 \end{cases}$$

Another variant of LBP is circular local binary pattern on P pixels with the same of radii R denoted as $LBP_{P,R}$. A local binary pattern that is rotation invariant is defined as

$$LBP^{rj}_{P,R} = \underset{j}{\text{minimum}}\ ROR(LBP_{P,R}, j),\ j = 0, \cdots, P-1 \tag{2.2}$$

The last descriptor is a PHOG which evaluates a pyramid histogram of oriented gradients. A binary edge image is first calculated using sobel edge detector. Then a pyramid is calculated by dividing the image at different levels. Finally the histogram is calculated by evaluating the descriptor at each level.

Tables 2.1 and 2.2 show a summary of the benchmarks available in the literature for both cervix segmentation and cervical cancer classification.

Table 2.1: Baseline results from different Segmentation benchmarks in the literature.

Authors	**Segmentation**
Denny et al. (2002) [26]	Direct Inspection
Kim et al. (2013) [24]	Optimized bounding box method
Song et al. (2015) [25]	Data driven approach using bounding box similarity

2.3.3 Segmentation and Region of Interest :General approaches

Segmentation describes the process of dividing the image into different regions using the uniformity of the colour pixels in the original image.Segmentation approaches could be divided into two general approaches: bounding box predictions and per pixel labeling. The bonding box prediction predicts a box with certain width, height and coordinates. We segment the image according to that bounding box. The per pixel segmentation gives each pixel in the image a certain label. The latter approach is considered more advanced and require much more complicated approaches. In

Table 2.2: Baseline results from different classification benchmarks in the literature.

Authors	**Samples**	**Classification**	**Sensitivity (%)**	**Specifity (%)**
Denny et al. (2002)[26]	2,754 cervigrams obtained from South Africa	Direct Inspection	70.00	79.00
Sankar et al. (2004) [27]	54,981 cervigrams from India and Africa	Screening	79.00	86.00
Kim and Huang (2013) [24]	2,000 images sampled from NIC	Majority vote	73.00	77.00
Song et al. (2015) [25]	280 images sampled from NIC	Multi modal with clustering	83.21	94.79
Xu et al. (2017) [28]	1,112 from NIC	Feature based CNN	80.9	75.9

this section we review some approaches for segmentation. These methods include approaches related to thresholding like Otsu [29] which searches for the number that decreases the standard deviation between the labels. Other methods by Adams and Bischof include region growing segmentation techniques like seeded region growing [30]. Edge detection approaches are also possible using convolutions with different operators like Sobel operators and Laplacian operators. Other approaches include clustering to gather similar colour channels together like in [31], graph based clustering [32] and fuzzy clustering [33].

CHAPTER 3

DEEP LEARNING APPROACHES

3.1 Introduction: Automatic Feature Extraction

An artificial neural network (ANN) is inspired by the biological neural network in animal brains. They are graph structures where the nodes are called neurons. Neurons take inputs and fire outputs using some activation functions. Neurons are connected together using edges. Edges contain real values called weights or parameters. These values are tuned using optimization algorithms to decrease the overall loss of the neural network. Neural networks work on input and output pairs (x, y) and fire outputs using forward propagation which is basically a composition of functions $\hat{y} = f_1(f_2(\cdots(f_n(x))\cdots))$. The value $\hat{y}$ represents the predicted output of the model. We are trying to minimize a loss function $g(y, \hat{y})$ using the parameters of the model.

Typically neural networks has the structure illustrated in Figure 3.1.

ANNs have the ability to extract features using the specified loss function. The

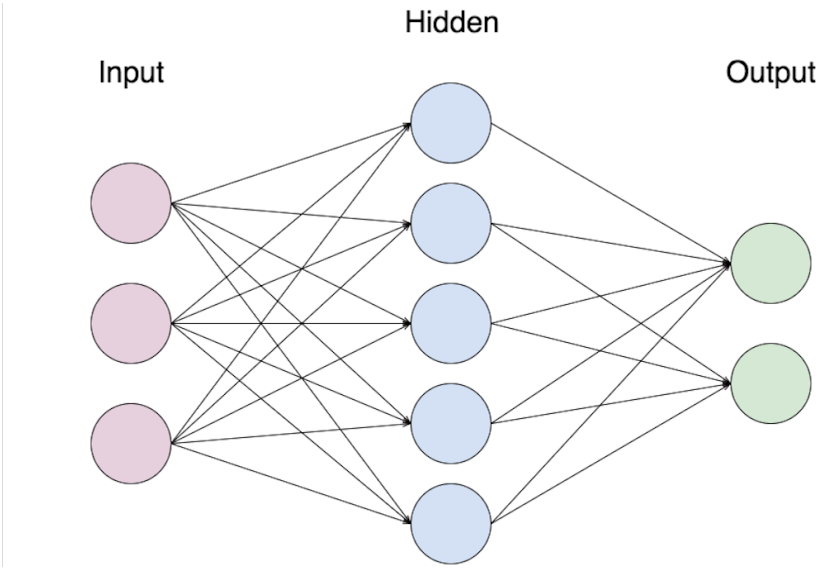

Figure 3.1: A neural network.

extracted features are stored abstractly as weights on the edges. Hence, when we take some inputs some neurons will cause a larger activation causing some features to prevail against others. Hence ANNs in general are considered as automatic feature extractors.

3.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks as in Figure 3.2 have gained a huge reputations after their success in the recent years especially in the ImageNet challenge which contains around 1 million images and to classify 1000 object types [34]. CNNs apply optimization methods and back-propagation in order to optimize the model's weights. Here is an overview of the main layers that a conventional CNN contains

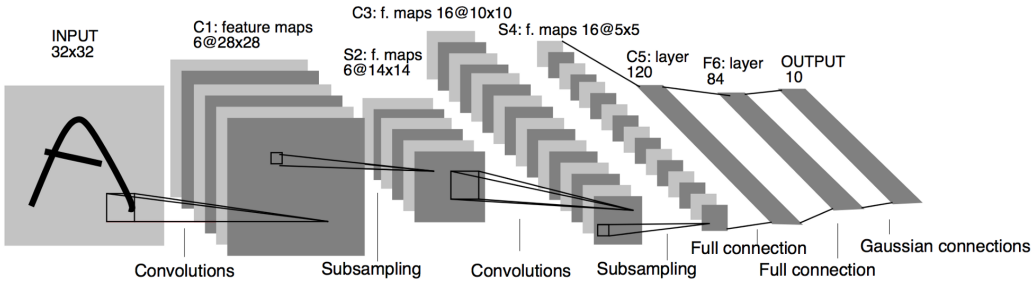

Figure 3.2: LeNet architecture by LeCun [1].

3.2.1 Convolutional layers

Convolutional layers are the main ingredients of CNNs. The main purpose of convolutional layers is applying convolutions with filters of fixed sizes. Convolution basically defines the operation of taking a locally weighted sum of the pixels at a certain window which is determined by the filter size. The **stride** S , defines how many pixels do we move at each time we apply the convolution. The **padding** P defines the number of zeros we attach to the width and height of the current input. Assume that we have an input size (H, W, C) where this corresponds to the height, width and number of channels respectively. Furthermore, F_w defines the width of the filter and F_h defines the height and N as the number of filters. Then, we define the dimension of the output as

$$H_o = \frac{H + 2P - F_h}{S} + 1$$

$$W_o = \frac{W + 2P - F_w}{S} + 1$$

Hence the output of the layer will be of dimension (H_o, W_o, N). Usually the number

of filters (depth) will increase as we progress along multiple convolutional layers and the dimension of the width and height will decrease because of applying pooling layers as in Figure 3.2.

3.2.2 Pooling layers

Pooling layers reduce the spatial size of the input. As we described earlier the number of filters will increase but we need to decrease the spatial size as we go deeper for otherwise we end up with millions of parameters. Furthermore, pooling layers help in providing the CNN with a zoomed view of the input by reducing the spatial dimension. Hence, the network will be able to recognize complex shapes across the whole input image. One of the most used types of pooling is the so called Max-pooling. A Max-pooling operation takes the window size and replaces it with the maximum number in the window. Typically, we take non-overlapping pooling operations across the input image. Usually we take these layers with shape $(2, 2)$ hence the spatial size of the input will decrease by factor of 2 in each dimension.

3.2.3 Non-linearities

There are different types of non-linearities that are applied to learn complex features and clip the output within a certain range. Here we discuss the most used ones.

The **sigmoid** or **logistic function** is one of the oldest types of non-linearities. A sigmoid function is defined as

$$g(z) = \frac{1}{1 + e^{-z}}$$

The sigmoid function can be thought of as a smoothed representation of a the non-differentiable step function. Its derivative take the form

$$g'(z) = g(z) \cdot (1 - g(z))$$

This function has the nice property of clipping the input to the range $(0, 1)$ which can be more or less interpreted as probability distribution function. The main disadvantages of the sigmoid function is that it is not zero centred and suffers from vanishing gradient problem when used in deep models [35].

The **tanh** non-linearity tries to resolve the problems of the sigmoid function. It can be defined in terms of exponents as

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

It has a smooth derivative and clips the input into the range $(-1, 1)$. As we see that the function is zero-centered.

The **rectified linear unit** (ReLU) is one of the most used functions in the deep learning architecture. It was part of the winning CNN in 2012 of the ImageNet challenge [34]. The function is defined as

$$f(z) = max(0, z)$$

Basically, the function clips inputs that are less than zero. Regardless of its simplicity and ease of computation it seems to work very well for deep networks [34].

3.2.4 Weight Regularization

Regularization is applied to reduce the risk of overfitting. Overfitting causes the model to become hooked to the training data and can't generalize to unseen data. As in conventional NNs we apply some restrictions on the parameters to control their behaviour. This also helps in reducing overfitting. There are mainly three types of constraints that we can add to the objective

L1 regularization

$$L(W; X) = f(W; X) + \lambda |W|_1$$

L2 regularization

$$L(W; X) = f(W; X) + \lambda |W|_2$$

Elastic net

$$L(W; X) = f(W; X) + \lambda_1 |W|_1 + \lambda_2 |W|_2$$

3.2.5 Dropout

Dropout as the name indicates deactivate some neurons in a certain layer as in Figure 3.3. To reduce overfitting we drop some of the neurons in training with some probability p. Hence we force the classifier to learn new features by deactivating some neurons [2].

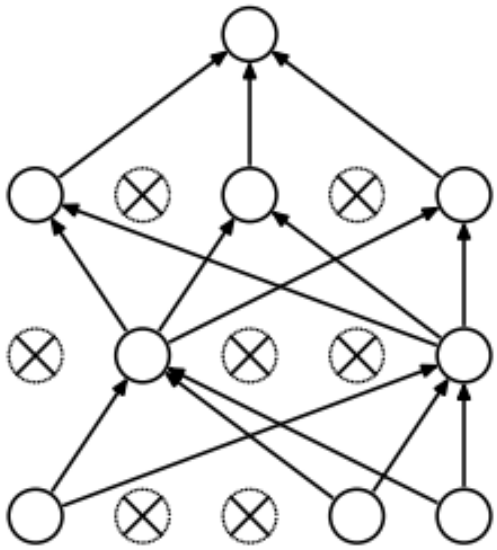

Figure 3.3: Dropout [2].

3.2.6 Classification Layers

The number of neurons in the first layer is equal to the output of the previous layer after unrolling the width, height and depth. For instance, assume that the shape of the final layer was (H, W, C) then we create the fist input layer of size $H \times W \times C$. Then the last layer will have the number of classes in the input.

On the other hand, global average pooling (GAP) helps in reducing the parameters

in the fully connected layers by taking the average over feature maps output of the previous layer. For instance suppose we have feature maps of size (H, W, C) from the previous layer, we then compute the average over each slice of size $H \times W$ hence the number of features will be of size $(1, C)$.

3.3 From CNNs to Deep Neural Networks (DNNs

3.3.1 Deep Networks

Deep learning had long history since the experiments of Hubel and Weisel on the visual cortex of cats [36]. They realized that some neurons on the cat's brain are stimulated by edges regardless of the position [36]. Since then, many attempts were implemented to model the visual cortex with code. Neurocognition was a layered structure with local receptive fields to activate a specific region at a time [37]. Then, back-propagation was used to train neural networks using a deep structure of convolutional layers called LeNet-5 [1]. In 2012 was the breakthrough of deep learning when AlexNet won the ImageNet challenge by 15.4 % top 5 error for classifying 1000 categories of around 1 million images with deep convolution neural network [34]. Top 5 error means that the image is classified correctly if one of the 5 highly scored classes contain the true class. AlexNet contained five conv layers concatenated with 2 classification layers with some pooling and normalization layers in addition to ReLU's non-linearities [34]. In 2013 ZFNet won the ImageNet challenge by 14.8 % with few changes to AlexNet by using bigger filters for convolutions [38]. In 2014 VGGNet was introduced with only using convolutional filters of size 3 by 3 with a model of size 16 [39]. VGGNet achieved 9.33 % top 5 error [39]. In the same year 2014 GoogLeNet achieved a lower error 9.13 % by introducing inception layers with less parameters than the previous architectures [40]. The inception layers contained different size convolution filters and max pooling with a

concatenation filter at the end [40]. In 2015 ResNet was introduced by Microsoft which achieved 6.7-5.7 % top 5 error on the ImageNet challenge [41]. The basic architecture contained 18-152 layers and a skip connection that passes some convolutional layers to compute the gradient without their existence [41]. ResNext was introduced in 2017 with top-5 error 5.6-5.3 % with the addition of residual layers.

3.3.2 Batch-Normalization layers

The BN-layer tries to solve the internal covariate shift which slows training and forces the network to highly depend on initial parameters [42]. At each stage of normalization we modify the mean and the variance of the current batch using the algorithm.

Algorithm 1 Batch Normalization

Input: Values of x over mini-batch $\mathcal{B} = \{x_{1\cdots m}\}$
Output: $\{y_i = \mathbf{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{m}^{i=1} x_i \tag{3.1}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \tag{3.2}$$

$$\hat{x_i} \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \tag{3.3}$$

$$y_i \leftarrow \gamma\hat{x_i} + \beta \tag{3.4}$$

Since we are applying smooth functions at each step of normalization the BN layers are differentiable and the error can be back-propagated through applying the chain rule.

3.3.3 Backpropagation and Update rule

Given a deep architecture we need a way to update the parameters of the model. Typically when we have a loss function with real value $f(W)$ where W are the parameters of the model that we need to minimize this function. Backpropagation basically moves back in the network trying to update the parameters using chain rule. For instance suppose we have only one variable in our network with three functions during feed-froward then we will have this structure

$$f(w) = f_1(f_2(f_3(x, w)))$$

where w is the network parameter and x is the input. We apply the squared difference as the loss

$$g(w) = (f(x, w) - y)^2$$

The the gradient will be back-propagated as

$$\frac{\partial g}{\partial w} = \frac{\partial g}{\partial f_3} \times \frac{\partial f_3}{\partial f_2} \times \frac{\partial f_2}{\partial f_1} \times \frac{\partial f_1}{\partial w}$$

Once we evaluate the gradient we need to update the parameters. We will need to change the weights to follow the gradient. For instance, the gradient descent optimizer updates the parameters like with learning rate α

$$w = w - \alpha \frac{\partial g}{\partial w}$$

3.4 Common Deep Learning Frameworks for Classification

3.4.1 Google Inception Models

Inception models are mini-modules that are contained in a bigger model. Basically there are two famous inception models. The first one is Inception-v3 discussed in [3]. It attempts to make faster predictions by reducing the number of parameters in the model. Typically, convolutions are factorized into smaller ones to decrease the parameters as illustrated in Figure 3.4.

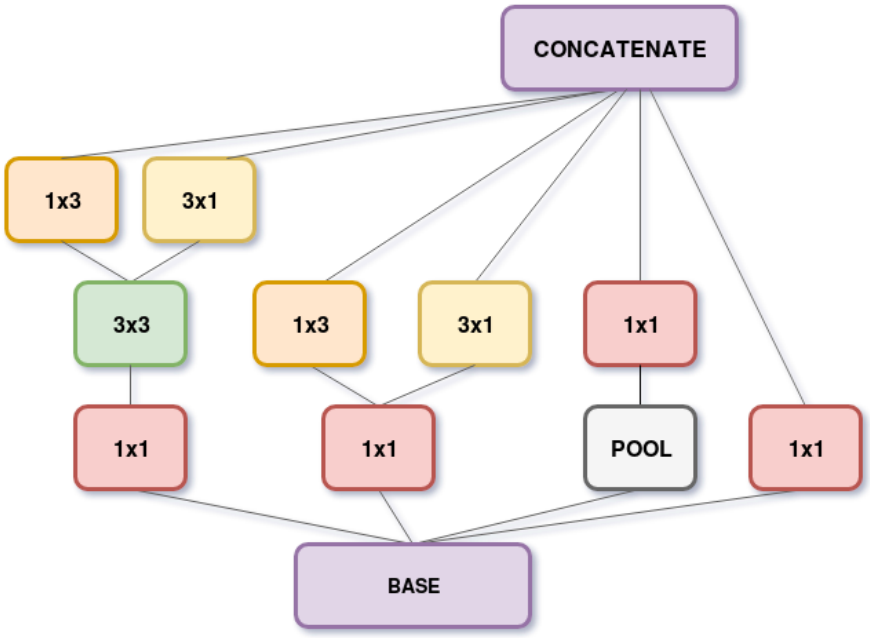

Figure 3.4: Inception v3 basic modules.

Such modules are repeated to construct the overall structure of the complete model

as in Figure 3.5.

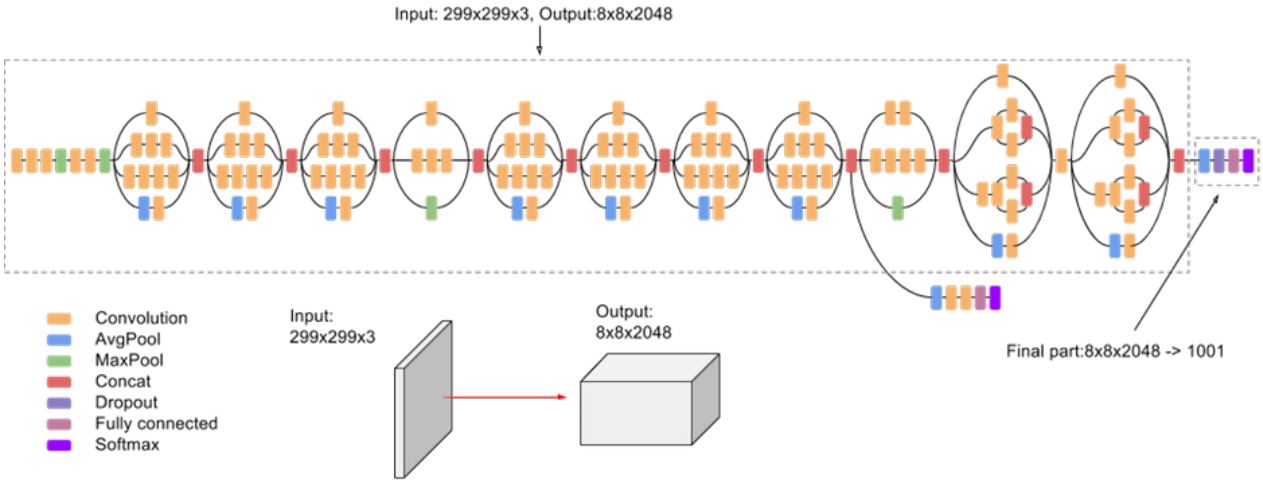

Figure 3.5: Inception v3 complete model [3].

Inception v4 builds on the previous version by adding residual layers. Residual layers are implemented by using skip connections that are used to jump over some layers in the model. Figure 3.6 shows the basic structures of the model.

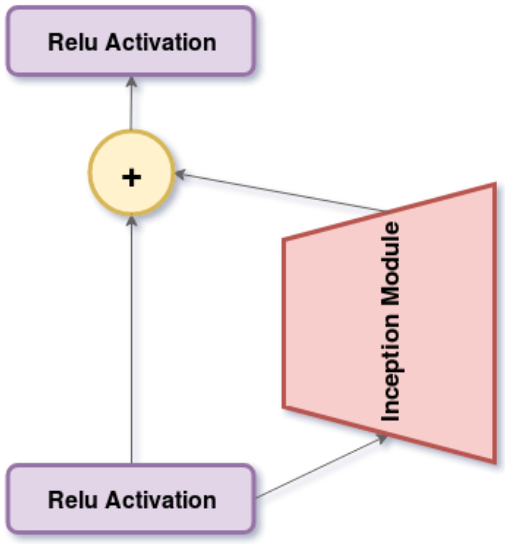

Figure 3.6: Inception v4 basic modules.

The over all architecture of the model is illustrated in Figure 3.7.

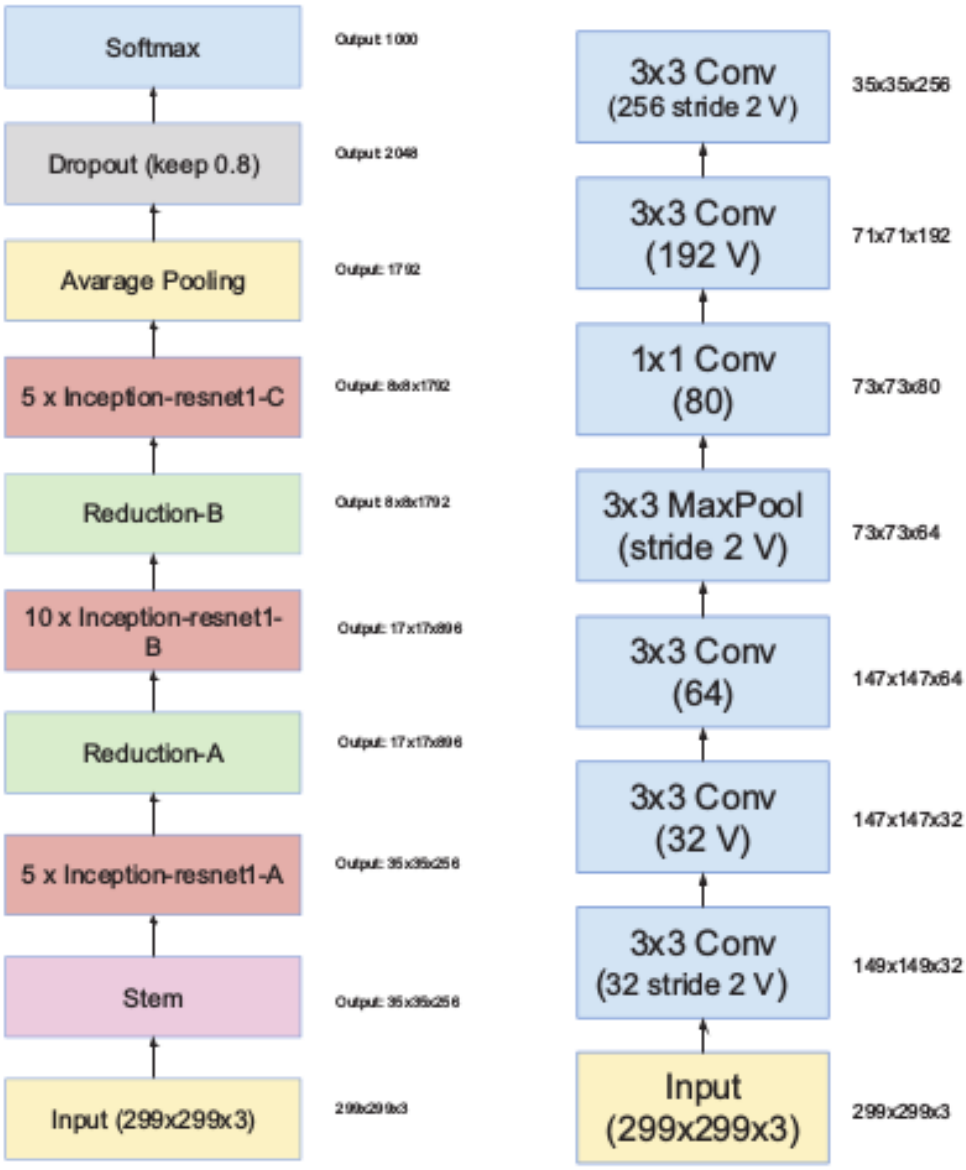

Figure 3.7: On the left is the overall structure and on the right is the inner layers of the stem [4].

3.5 Common Deep Learning Segmentation and Detection Models

3.5.1 Evaluation Metrics

Intersection over union (IoU) is usually used which is evaluating how accurate the bounding box is. Given b_{true} and b_{pred} boxes we calculated the metric as

$$\textbf{IoU} = \frac{b_{\text{true}} \cap b_{\text{pred}}}{b_{\text{true}} \cup b_{\text{pred}}}$$

Since intersection is always bigger than union, the value is always between 0 and 1. The closer the value to 1 the better the accuracy. For images in the validation set we average the predicted and true bounding boxes over all the images

$$\textbf{Validation Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \textbf{IoU}_i$$

Another metric that is used for both segmentation and detection is mean average precision **mAP** which is the average precision computed over all the classes.

3.5.2 Object Detection Models

Typically we want to detect a bounding box around the object that we want to detect. The bounding box should enclose the whole body with boundaries of the object as

tight as possible to the edges of the box. The models will predict the corners the bounding box.

Rich feature hierarchies for accurate object detection and semantic segmentation (R-CNN) was developed in 2014 [43]. The basic idea is to use selective search [44]. Selective search avoids exhaustive search by starting by small bounding boxes all over the image and the groups them in a hierarchical format. They are grouped using the color features and some similarity metrics. R-CNN networks combines the selective search and deep network features to reduce the number of region proposals for each detected object. a 4096-dimensional vector is extracted for each region proposal using a pretrained alexNet model which contains five conv layers and two classification layers. The feature vector is then inserted into multiple classification layers to result in a probability distribution of the classes. There is also another regression branch to reduce the localization error of the bounding boxes. The model achieves 30% better **mAP** score than the previous models with 53.3% **mAP** score over the VOC 2012 dataset.

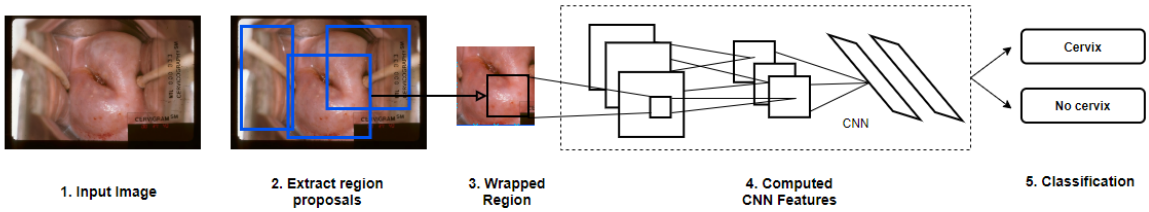

Figure 3.8: R-CNN architecture.

The model was improved using fast R-CNN model [45]. The network inputs the image with multiple convolutional and max pooling layers. After that, a region of

interest is extracted for every region proposal. Each feature vector is inserted into a multiple classification layers that are divided into two networks: one to produce class probabilities and the other is a regression network that produces bounding box predictions. This allows for training on a single stage with multiple task loss, the training step can update all the layers and no storage is needed on the disk for caching the features. This makes the network 10 times faster for training than R-CNN while achieving a better **mAP** score.

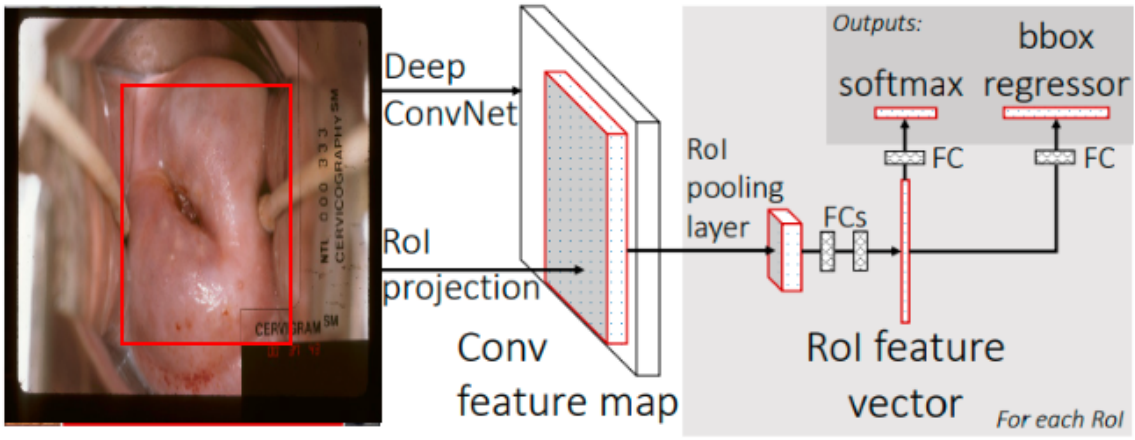

Figure 3.9: Fast R-CNN architecture.

Faster R-CNN [46] improves the Fast R-CNN model by getting rid of the selective search method which is computational expensive. The model uses region proposal network (RPN) to extract the region proposals, detect the objects and predict the bounding boxes.

A model takes the full image is an input. A sliding window of shape 3×3 is used to output a features vector linked to two classification layers. The first one is for regression and the second one is for classification. Many proposals are created by

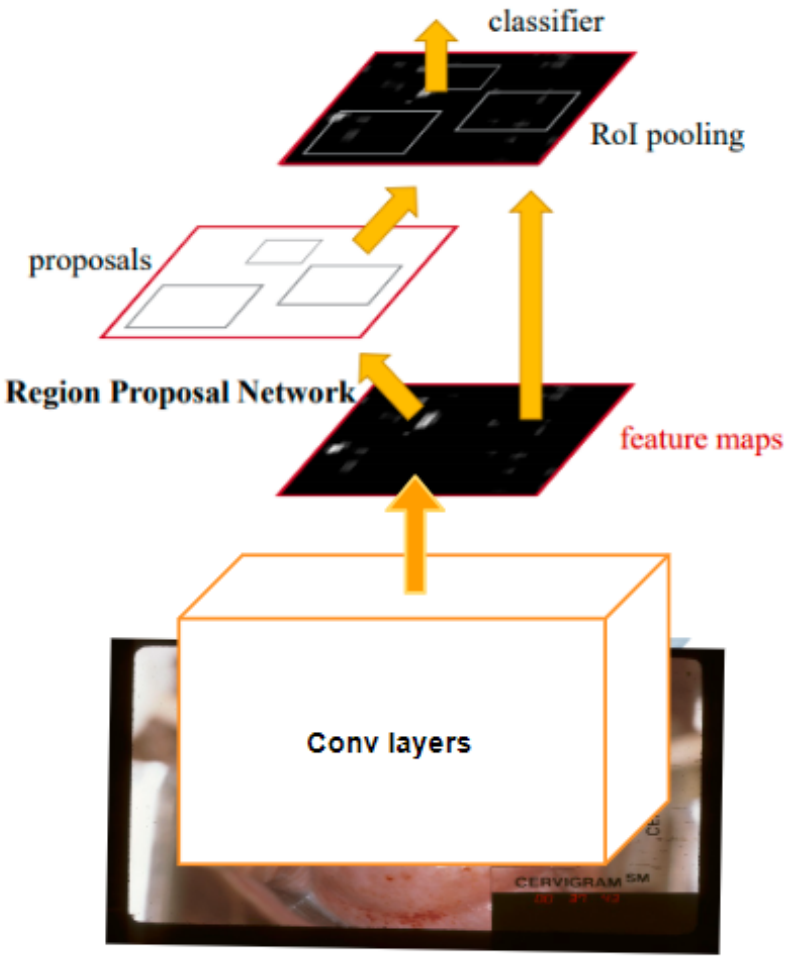

Figure 3.10: Faster R-CNN architecture.

these classification layers. If we extract k regions then the output of the regression layers has a shape of $4k$ (the coordinates, the widths and the heights) and the output shape of the classification layer is $2k$ (object / or no object). These k bounding boxes are called anchors.

3.5.3 Object Segmentation Models

In segmentation we give each pixel in the image a certain class. For instance, in cervix segmentation the semantic map will contain two labels; the cervix and the background

as in Figure 3.11.

Figure 3.11: An example of segmenting a cervix.

Basically there are two types of segmentation. The first one is semantic segmentation, which basically extracts the masks of the objects in the image. Semantic segmentation does not care if there are many occurrences of the same object in the same image as it gives them the same label. On the other hand, instance-level segmentation also segments the different occurrences of the same object in the image.

U-Net is a convolutional neural networks that was originally designed to work for biomedical images [5]. The architecture consists of a "contracting path" and an "expansive path". The "contracting path" meets the typical design of a convolutional network. It constitutes some 3 by 3 convolutions, each are connected to an activation function and a pooling operation for downsampling. Every step in the "expansive path" consists of an transposed operation of convolution that increases the number of activation by a factor of 2, a concatenation with the correspondingly cropped feature map from the contracting path, and some 3 by 3 conv layers, each inserted into an activation function. Finally, there is a layer of shape 1 by 1 convolution that maps each 64 vector map to the number of classes we desire. Figure 3.12 shows the overall

architecture of the U-Net model.

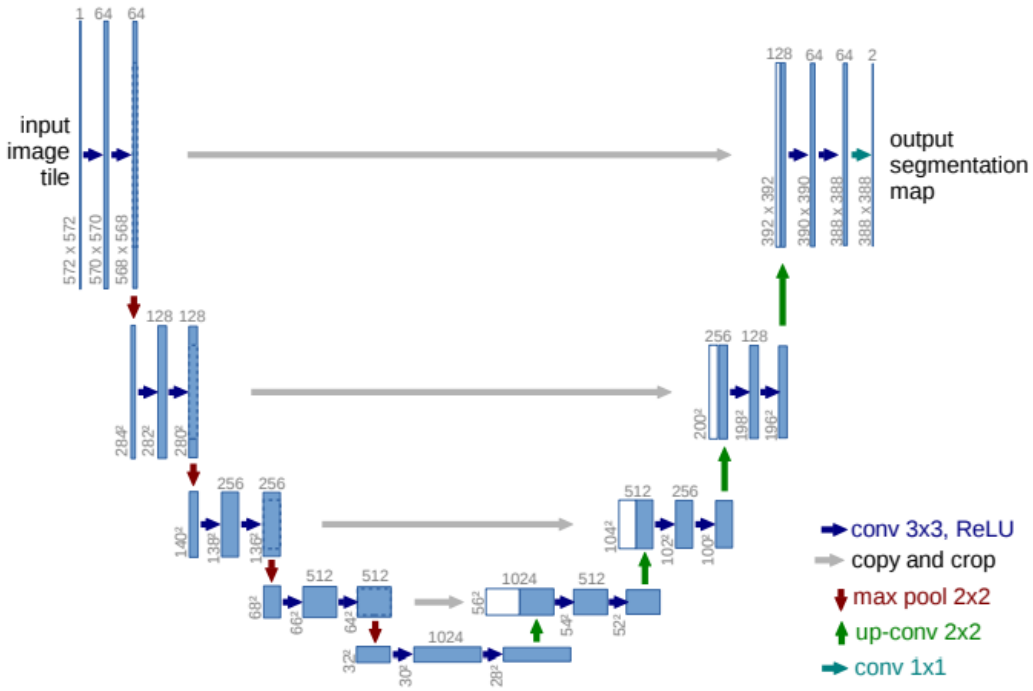

Figure 3.12: The architecture of a U-Net model [5].

The SegNet architecture is a similar model to U-Net but can be generally used to a diverse number of tasks [6]. SegNet follows the architecture of encoder-decoder networks. The encoder network consists of 13 layers that follow the same architecture of the VGG network [39]. Each encoder consists of convolutional layers, batch normalization layers, rectified linear units (ReLU) and max-polling to down-sample the activation maps. Since we are using multiple max pooling layers this results in increasingly lossy (boundary detail) image representation. Hence memorizing the indices of the max-pooling for each feature map in the encoder. The decoder network contains the same number of layers as in the encoder. It uses up-sampling using the memorized max-pooling indices from the corresponding max-pooling layers indices.

The output of the decoder is fed into a multi-class soft-max classifier to generate the class probabilities independently for each pixel.

Figure 3.13 shows the overall architecture of the SegNet model.

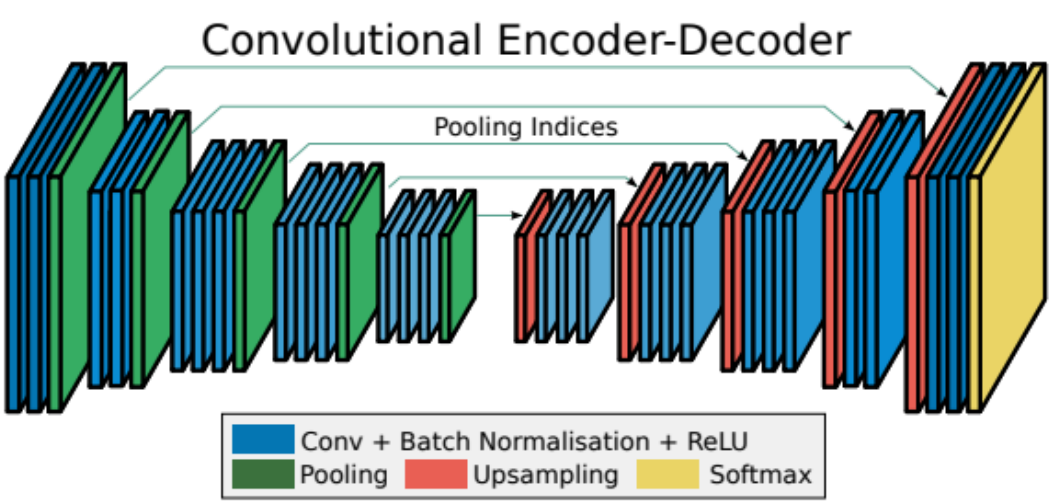

Figure 3.13: The architecture of a SegNet model [6].

Mask R-CNN [7] extends Faster R-CNN by adding a third branch that outputs the object mask. Mask R-CNN uses the same first stage in the Fast R-CNN architecture which is the region proposal network (RPN). In parallel with predicting the box offset and the class label it outputs a binary mask for each region of interest (RoI). This masks the mask predictions independent of the class predictions. For each sampled RoI a multi-task loss is applied as

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}$$

The mask has Km^2 shape for each region of interest where we have K masks each of size $m \times m$. A per-pixel sigmoid is applied so L_{mas} is the average binary cross-entropy loss. The basic architecture of Mask R-CNN contains two parts. The first one is a convolutional backbone architecture that is mainly used for feature extraction. Different models are tested for the backbone architecture like ResNet and ResNext. The second part is the head for classification, bounding box prediction and mask prediction for each defined RoI. Figure 3.14 shows the overall architecture of the Mask R-CNN model.

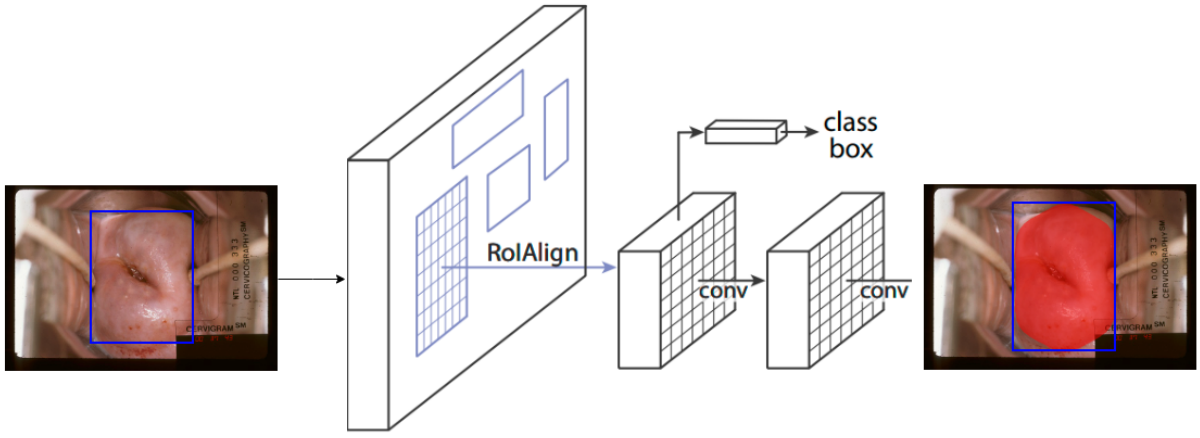

Figure 3.14: The architecture of a Mask R-CNN model [7].

You only look once (YOLO) is an object detection approach that uses full evaluation of the image to extract the bounding boxes and their corresponding probabilities. YOLO processes images in real-time approximately 45 frames per second achieving state of the art performance in object detection in real time [8]. YOLO uses one network and information from the full image to output each prediction. The image is first divided into $S \times S$ grid with equal area of each grid. Every grid will then

predict B bounding boxes and their corresponding confidence scores. The confidence score gives an impression about the model confidence of an existing object inside the bounding box. The confidence is defined as

$$\Pr(\text{Object}) \times \text{IOU}_{\text{pred}}^{\text{truth}}$$

where IOU defines the intersection over union metric. Given two bounding boxes b_1, b_2 IOU is defined as

$$\text{IOU}_{b_2}^{b_1} = \frac{b_1 \cap b_2}{b_1 \cup b_2}$$

In short, it evaluates the overlapped area of the bounding boxes with respect to their union. This metric takes values in the interval $(0, 1]$ where the higher means the better.

Each bounding box is associated with 5 parameters x, y, w, h and confidence. Each grid cell evaluates C conditional probabilities $\Pr(\text{Class}_i|\text{Object})$. Then class specific probabilities are evaluated using

$$\Pr(\text{Class}_i|\text{Object}) \times \Pr(\text{Object}) \times \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) \times \text{IOU}_{\text{pred}}^{\text{truth}}$$

The parameters of the bounding box predictions are encoded as $S \times S \times (5B + C)$.

The network architecture is modified version of GoogLeNet for image classification with 24 conv layers connected to 2 FC layers. There are Maxpool layers to reduce the spatial size of the features at each level of the network.

The Basic architecture of the CNN is shown in Figure 3.15.

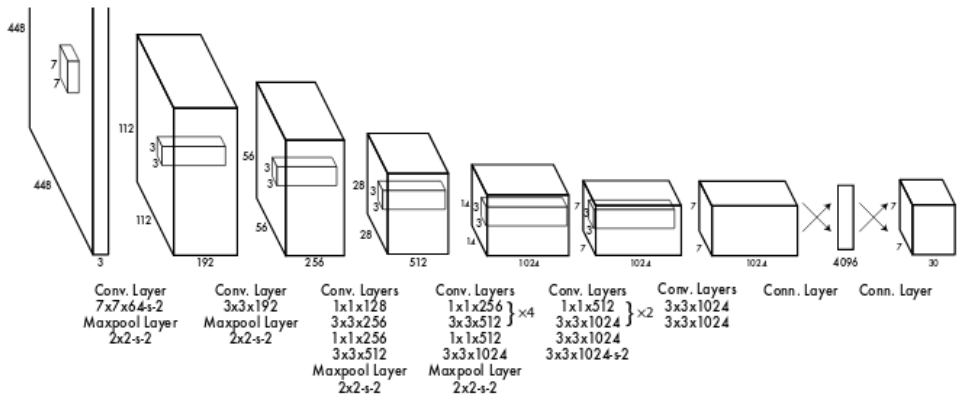

Figure 3.15: YOLO architecture [8].

The loss function, according to [8], evaluates the sum squared error between the truth value and the prediction. λ parameters to give loss of localization more value than the loss related to classification.

$$Loss = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \tag{3.5}$$

$$+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \tag{3.6}$$

$$+\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2 \tag{3.7}$$

$$+\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \tag{3.8}$$

CHAPTER 4

FULLY-AUTOMATED DEEP LEARNING PIPELINE

4.1 Introduction

This chapter describes the general structure of our pipeleine. We provide a fully automated approach with multiple structures with each structure as a phase of the pipeline. The pipeline starts by segmentation which extracts a bounding box prediction of the images in the dataset. Then we apply preprocessing of the dataset. We then apply augmentation and feature extraction using hand-crafted features and convolutional neural networks. Finally, we use a classifier to extract the labels. Note that this approach is fully automated with no assistance from a physician or outside interference. Figure 4.1 shows the general pipeline of our model.

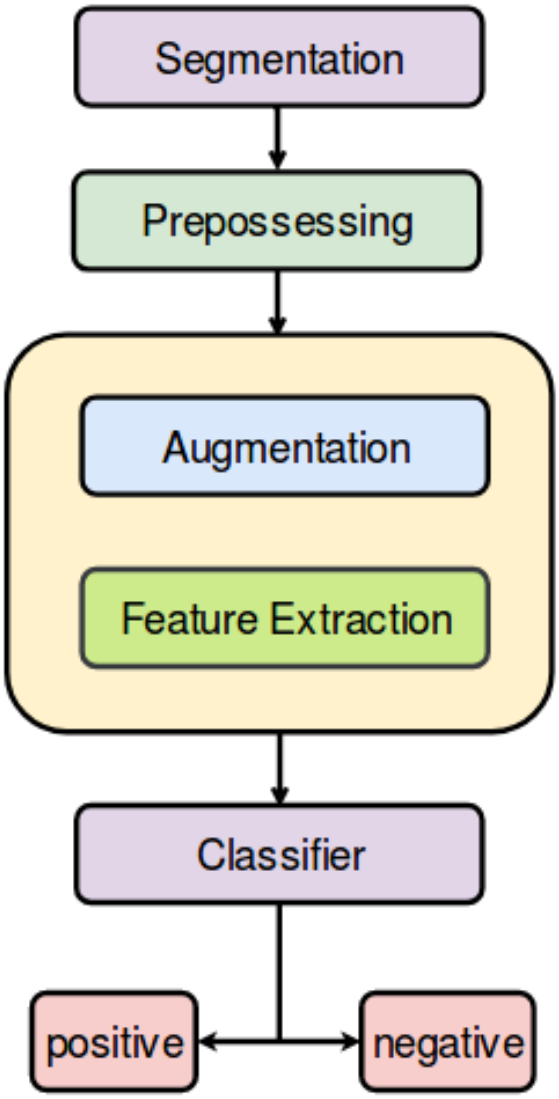

Figure 4.1: The pipeline of our architecture.

4.2 Proposed Segmentation Approach

We used you only look once (YOLO) [47] for segmenting the images in the dataset using bounding box prediction. The general approach takes the following stages for generating the bounding boxes for all the images in the dataset.

4.2.1 Data Preparation

We used around 1500 images with bounding boxes annotated with an expert in medicine taken from [48]. We use the annotated data for training the YOLO network explained in the previous chapter. We ignore the classification module for YOLO because we only need to predict the bounding boxes. The annotated images are associated with csv file which contains the bounding box parameters x, y, w, h, where (x, y) defines the coordinate of the top left corner of the bounding box and (w, h) define the width and the height of the box. In the other hand ,YOLO accepts 5 parameters: **class** which defines the label of the current image, $\mathbf{center}_x$ as the relative x-center of the box, $\mathbf{center}_y$ as the relative y-center of the box, $\mathbf{relative}_w$ as the relative width of the box with respect to the width of the image and $\mathbf{relative}_h$ as the relative height. Hence, given x, y, w, h for the bounding box and W, H where W is the width of the image and H is the height for the image, the parameters could be evaluated as

$$\text{center}_x = \frac{x + w/2}{W}, \ \text{center}_y = \frac{y + h/2}{H}$$

$$\text{relative}_w = \frac{w}{W}, \ \text{relative}_h = \frac{h}{H}$$

See Figure 4.2 for visualization.

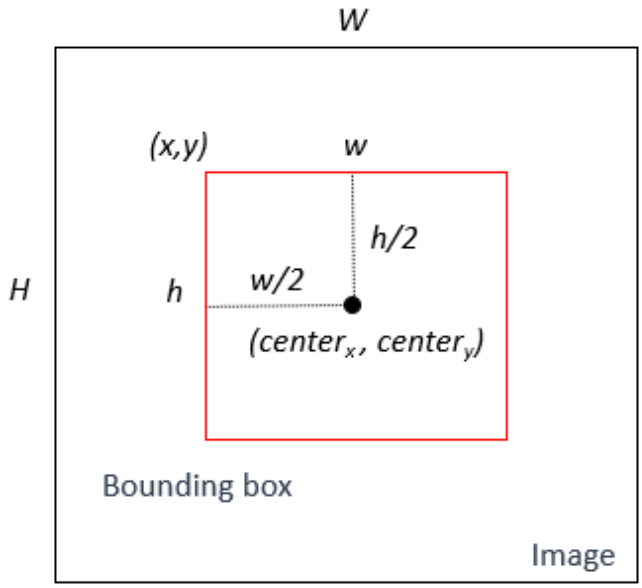

Figure 4.2: Visualization of the bounding box parameters.

4.2.2 Performance Measure

We use intersection over union to evaluate our segmentation method

$$\text{IOU}_{b_2}^{b_1} = \frac{b_1 \cap b_2}{b_1 \cup b_2}$$

where $b1$ is the true bounding box (as marked by a specialist in the medical field) and $b2$ is the predicted bounding box from the YOLO neural network. For validation we evaluate the average intersection over union metric of all the images in the validation set.

$$\frac{1}{N}\sum_{i=1}^{N} \mathrm{IOU}_{\mathrm{pred}_i}^{\mathrm{truth}_i}$$

Training and Validation

We used pretrained YOLO model (trained on COCO dataset [49] for training the detection network. The annotated data is first divided into 80% for training and 20% for validation. Table 4.1 shows the parameters we chose to train YOLO where **batch size** defines the number of images to train at a time, **subdivisions** further divides the batch size into divisions, **momentum** defines the acceleration of the optimization algorithm, **learning rate** defines how fast we update the parameters of the model, **conv layers** is the number of the convolutional layers, **FC** defines the number of the fully connected layers.

Table 4.1: Yolo parameters

Batch Size	64
Subdivisions	4
Momentum	0.9
Decay	0.0005
Learning rate	0.0001
Conv layers	22
FC layers	2
Classes	1

4.3 Proposed Classification Approach

4.3.1 Preprocessing

Before using the classification model we need to preprocess the dataset. Normalization and zero-centering are popular approaches [50]. The input to our images with the following shape (H, W, C) where H is the height of the image , W is the width of the image and C is the number of channels which is 3 if the image is colored and 1 if the image is gray. We evaluate the mean and variance along each channel. Then for each pixel in our dataset we normalize the pixel value according to its channel

$$\frac{x - m_c}{v_c}$$

where $c \in [1, 2, 3]$ corresponding to each channel where m_c, v_c are the mean and variance of the corresponding channel . After that, we crop our images according to the YOLO segmentation network. We also scale the images to size 256×256 to make the images accessible by most classification models.

4.3.2 Performance Measure

Performance measure

Given a true label and a predicted label from a classifier we show in Table 4.2 the relations between the true classes and predicted classes. We use Sensitivity or True

		True Class	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 4.2: True positive, true negative, false positive and false negative.

Positive Rate (TPR) which calculates the percentage of the positive classes correctly classified by the model

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.1}$$

Specificity or True Negative Rate (TNR) which calculates the percentage of the negative images correctly classified by the model

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.2}$$

and accuracy (ACC) which is a combination of the previous metrics

$$\text{ACC} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{TP} + \text{FN}} \tag{4.3}$$

We report the three metrics and compare the results to engineered features classification using different descriptors. We also report receiver operating characteristic curve (ROC) which calculates the performance of a binary classifier as we vary the classification threshold in the interval $(0, 1)$. The threshold defines the value that differentiates the negative and positive classes. Given a value a for the threshold if the

predicted value from the neural network is p then if $p < a$ we classify it as a negative and if $p \geq a$ we classify it as the positive class. Furthermore, we use the area under the curve (AUC) metric which calculates the area under the ROC curve. Figure 4.3 is a visualization of ROC and AUC.

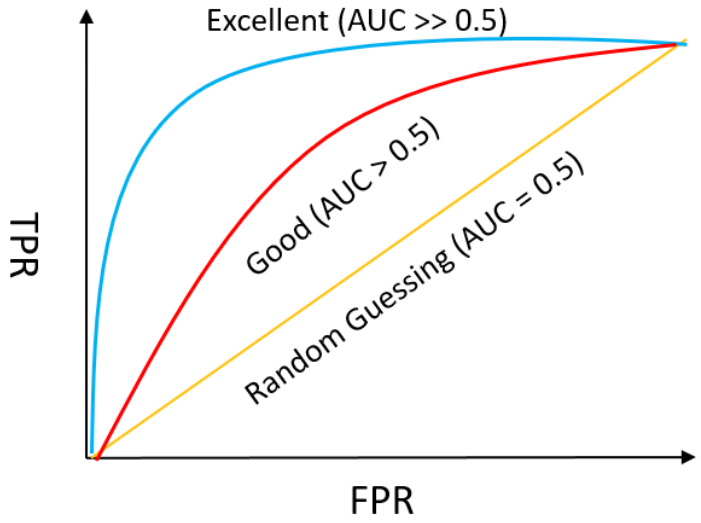

Figure 4.3: Visualization the ROC graph and AUC values for random, good and excellent classifiers.

4.3.3 Training and Validation

We use 10-fold cross validation to compare the accuracy of different models. We run the classification model for 10 iterations. At each iteration we hold 10 % of the data for validation. In order to avoid having different number of samples for each class we use stratified sampling were we make sure that each fold contains a balanced number of classes. At the end of the 10 iterations we evaluate both the mean and standard deviation of the metrics of the accuracy, specificity and sensitivity.

Algorithm 2 Cross Validation

Input: data, model

$i = 0$

while $i < 10$ **do**

$train, valid = getFold(data, i)$;

$model.fit(train)$;

$h \leftarrow model.predict(valid)$;

$i \leftarrow i + 1$;

$reportMetrics(h)$;

4.3.4 Hand-crafted Features

We use three types of hand-crafated features. PLAB, calculates a pyramid of histograms in the L*a*b* space. The L*a*b space is a 3-axis color system with dimension L for the luminance and a and b for the color dimensions as illustrated in Figure 4.4.

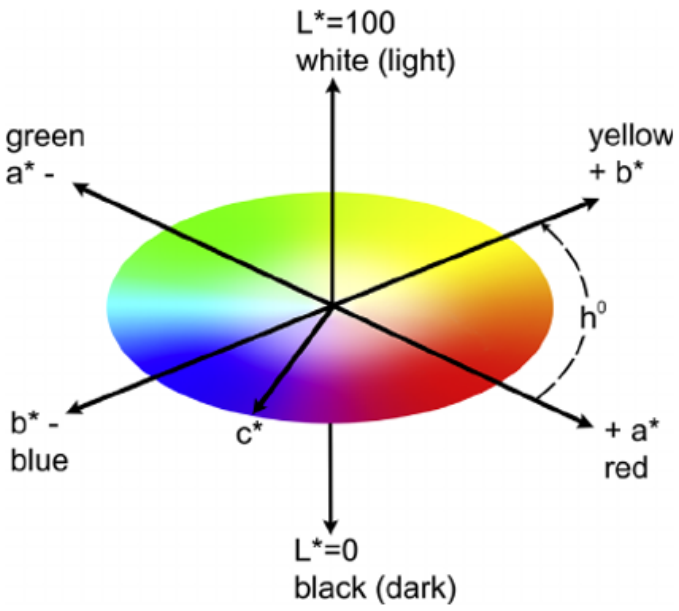

Figure 4.4: L*a*b* color space.

The **Histogram** is an estimate of the probability distribution of a continuous variable (quantitative variable) using ranges of values called bins. In this task the

histogram is a bar graph where the height represents the number of the pixels in a certain range. The ranges are evaluated by using the min and max values of the pixels and we divide that into equal regions where each region is called the bin of the histogram. See Figure 4.5 for an example of a histogram.

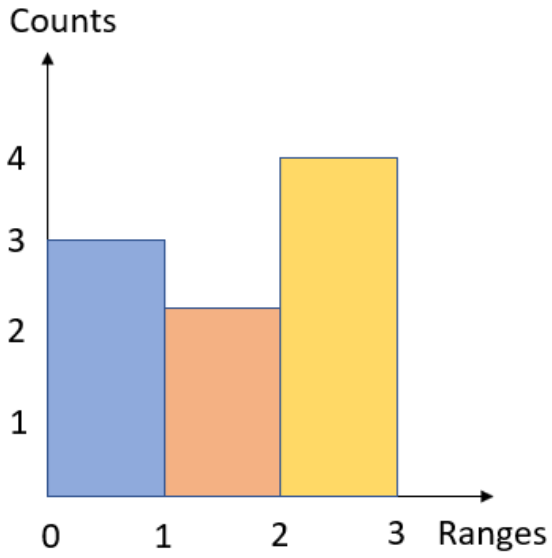

Figure 4.5: An example of a histogram for the the set of values (2.3, 3, 2.5, 0, 0.5, 1.5, 2, 0.3, 2.9).

We extract three pyrmaids by dividing the region into three different levels. The first level contains 1 region, the second level contains 4 regions and the last level contains 16 regions. The bin size for the histogram is fixed to be 16; hence this creates a descriptor of size $(16 + 4 \times 16 + 16 \times 16) \times 3 = 1008$. This is illustrated in Figure 4.6.

The second used descriptor is a pyramid of locally binary patterns (PLBP). The local binary patter is evaluated as the following. At each pixel located at (x_c, y_c) calculate the local binary pattern

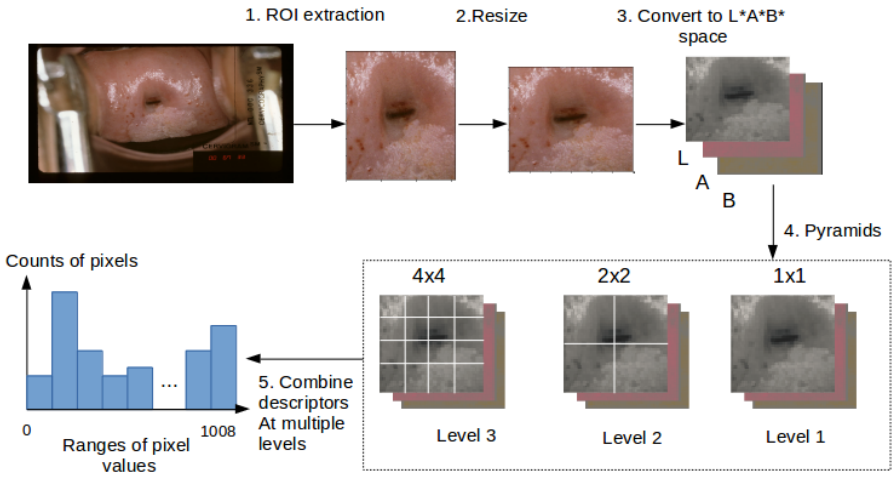

Figure 4.6: PLAB descriptor.

$$LBP(x_c, y_c) = \sum_{p=0}^{7} s(i_p - i_c)2^p \tag{4.4}$$

where i_c is the middle pixel value and i_p corresponds to the gray scale value of the neighbor pixel. Note that each center pixel where have 8 neighbors $c \in \{0, 1, \cdots, 7\}$. Given x is the subtracted pixel values we define $s(x)$ the a sign function as

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

In Figure 4.7 we illustrate a simple example of the evaluation of LBP for the center pixel.

Another variant of LBP is circular local binary pattern on P pixels with the same of radii R denoted as $LBP_{P,R}$. A local binary pattern that is rotation invariant is defined as

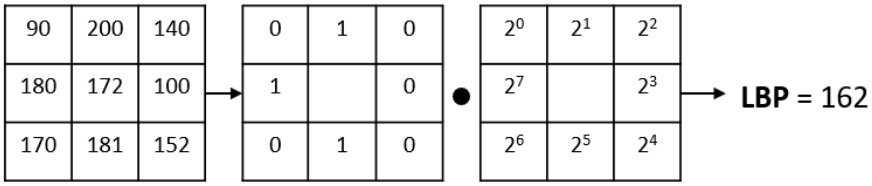

Figure 4.7: LBP procedure.

$$LBP^{rj}_{P,R} = \underset{j}{\text{minimum}}\ ROR(LBP_{P,R}, j),\ \ j = 0, \cdots, P-1 \tag{4.5}$$

We used the circular local binary pattern descriptor with fixed parameters $P = 8$ which defines the number of equally spaced pixels to the middle pixel and the radius is fixed to be $R = 1$. The descriptor was evaluated on the gray scale image by dividing the image into three regions of size 1, 4, 16 and 64 respectively. The number of bins is 10 at each region. This resulted in a descriptor of size $10+4\times10+16\times10+64\times10 = 850$ feature size.

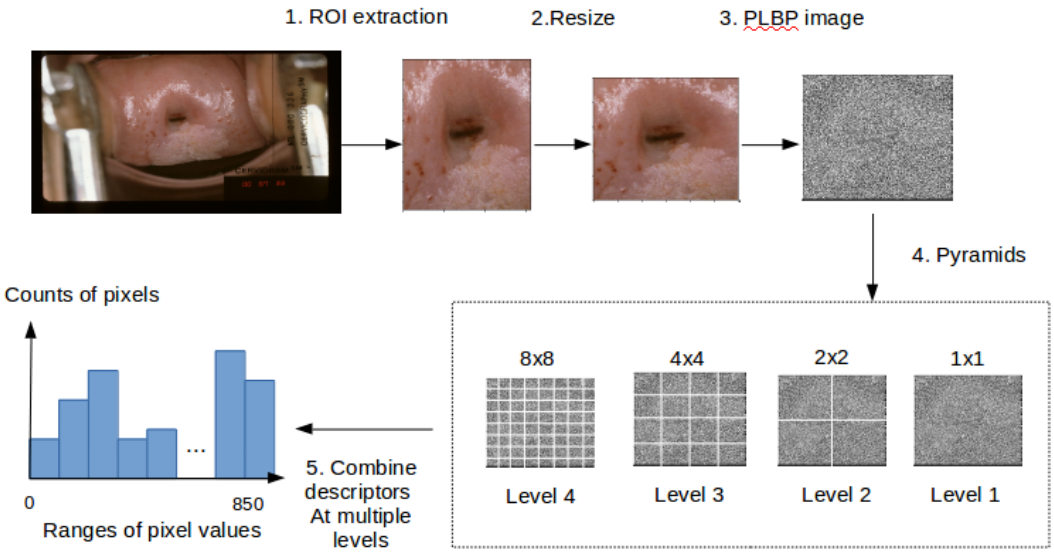

Figure 4.8: PLBP descriptor.

The last descriptor is a PHOG which evaluates a pyramid histogram of oriented gradients. A binary edge image is first calculated using sobel edge detector. The sobel edge detector extracts the edges or boundaries of the objects in the images. The levels of the pyramid are 1, 4, 16 and 64. It is tested that 8 is the best bin size. This results in a descriptor of size $8 + 4 \times 8 + 16 \times 8 + 64 \times 8 = 680$.

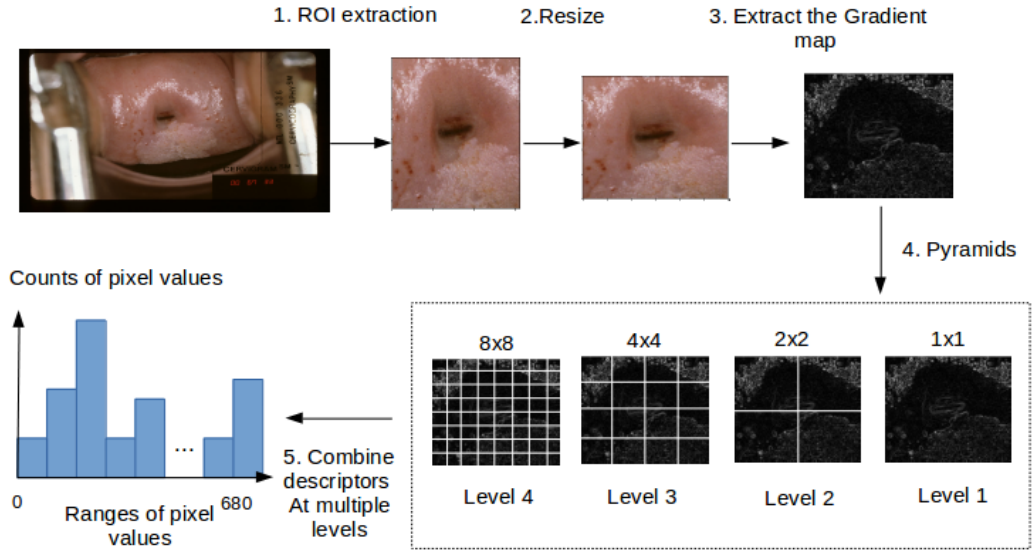

Figure 4.9: PHOG descriptor.

Hence by concatenating the three descriptors we obtain a vector of size 1008+850+680 = 2538. The descriptor vector is then fed into a neural network for classification.

4.3.5 CNN Features

These features are automatically extracted from the images using a convolutional neural network. We first extract the region of interest from the image, then resize the image. We usually call matrices $\mathbb{R}^{n_1 n_2 \cdots n_k}$ of high dimensions "tensors". We resize the images to tensors of the shape $(256, 256, 3)$ where the first and second components

are the width and the heights of the image respectively and the last component is the number of channels which 3 corresponding to the RGB color space. We feed into the CNN tensors of images of shapes $(N, 256, 256, 3)$ where N is the number of images we train at a time. We apply a couple of convolutional layers and max-pooling layers to extract the features at each depth of the neural network. The output of these operations create tensors of the shape (N, w, h, M) where (w, h) are the truncated width and heights of the image as a result of applying the pooling which decreases the spatial size of the input by half. The last dimension M is the number of the filters at the last convolutional layer. In order to feed these array into the classifier module we first need to reshape these arrays. This is a simple operation where we broadcast the last three dimensions into one dimension and create an output of the shape $(N, w \times h \times M)$. Usually this operation is called "flattening".

Figures 4.10 and 4.11 show CNN features extracted from different models

4.3.6 Classifier

The classifier is a basic neural network with inputs as tensors of the shape (N, L) and the output are tensors that have the shape $(N, 1)$ where N is the number of the feature vectors extracted from the hand-crafted descriptors of the convolutional neural network at each time we are training and L is the size of the extracted feature vector. In order to train a binary classifier that classifies the images into positive or negative we must convert the output into a probability distribution i.e the output value at each

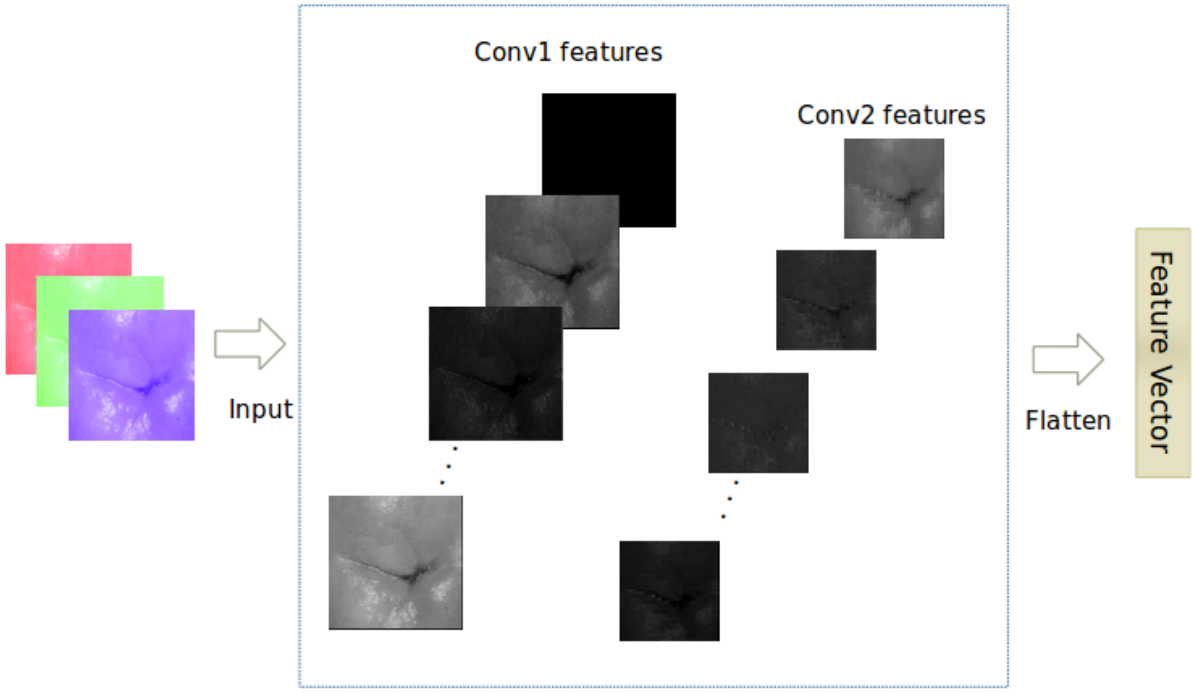

Figure 4.10: CNN features extracted from a model with two conv layers.

time should be between $(0, 1)$. Hence we use the sigmoid function defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

where x is the output of the neural network. The sigmoid function maps each feature vector to the range $(0, 1)$. The loss function calculates how good we are training our model by comparing the true label to the predicted probability. We need this value to be minimized. It is evaluated using a binary loss classifier

$$Loss = -\left(y \log(p) + (1 - y) \log(1 - p)\right)$$

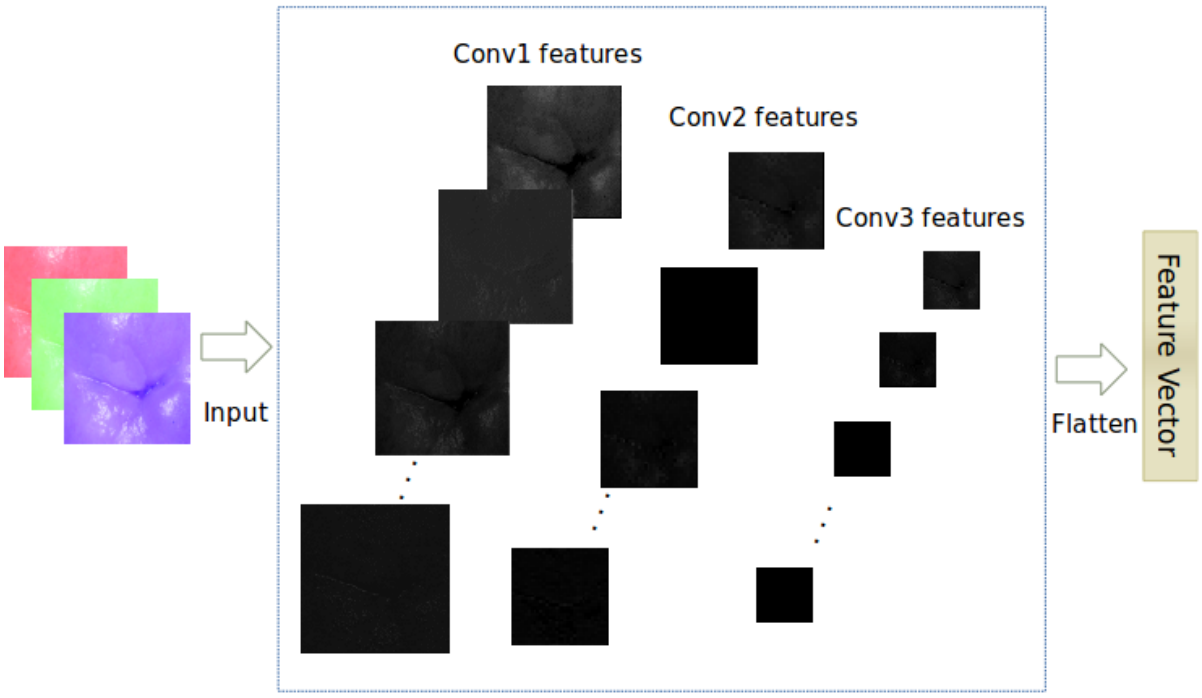

Figure 4.11: CNN features extracted from a model with three conv layers.

where y is the truth value which takes either 0 or 1 and p is the probability of the class that is evaluated using the sigmoid function. Figure 4.12 shows the procedure of mapping features to a probability distribution using a neural network.

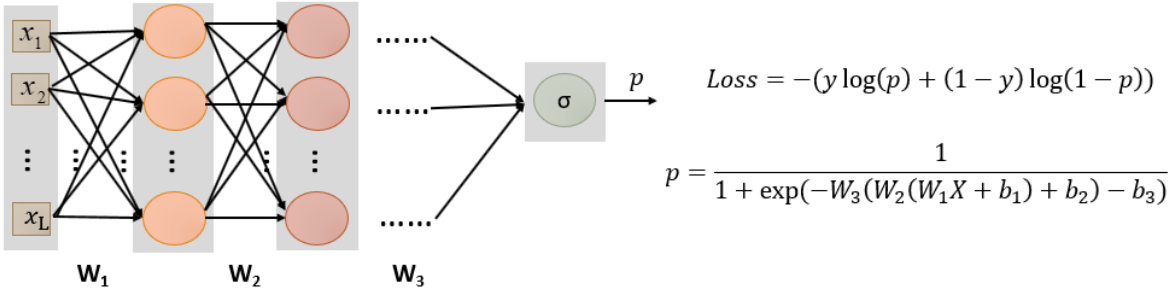

Figure 4.12: A binary classifier.

CHAPTER 5

EVALUATION OF THE PROPOSED APPROACH

5.1 System Configuration

The segmentation model was trained on "you only look once" (YOLOv2) on a machine with Nvidia Quadro M4000 card with 8GB memory card and 16.04 Linux Ubuntu version. The system was configured to work with Cuda 8.0 and cuDDN to make the training process faster.

The classification experiments are done on a Device with Nvidia GeForce GT 740M with 2GB memory. We used Keras with TensorFlow backend for creating the CNN models. The models were trained on a 16.04 Linux Ubuntu version. We also used Google Colaboratory for some of the experiments. It offers 8GB of free GPU memory for research purposes.

5.2 Dataset

5.2.1 Overview

The dataset was extracted from a study by a large medical data archive gathered by the National Cancer Institute (NCI) in the Guanacaste project [51]. The data consists of images from around 7,000 patient visits with around 44,000 cervigrams. The dataset contains relevant information like age, worst histology, HPV status, cervigram and number of days between the collection of the images and the histology test. Table 5.1 shows a description of of the fields available in the dataset along with the possible values for each field. We note that some fields have negative values which indicate the absence of the values for that particular field. We had to take that into account especially when collecting the data for our experiments.

Figure 5.1 shows a comparison of the fields that we described earlier. We can infer lots of information from that figure. We can infer from the middle graph which calculates the counts of each class of the histology analysis cases that most of the data is not labeled , i.e it has a label = -2. The graphs of the worst histology and HPV status shows that there is a correlation between them as the HPV status increases there is an increase risk of cancer. From the graphs there is no clear correlation between the age and either worst histology or HPV.

Figure 5.2 shows a comparion between the worst histology and HPV status.

Table 5.1: Description of the fields in the dataset.

Field	**Description**	**Possible Values**
AGE GRP	Age of the patient at the image collection date divided in groups of 5.	• -1 = Uknown • 1 = < 20 • 2 = 20-24 • 3 = 25-29 • 4 = 30-34
WRST HIST AFTER	Worst Histology analysis on or after the image collection	• -2 = No histology • 0 = Normal • 1 = CIN1 • 2 = CIN2 • 3 = CIN3 • 4 = Cancer
HPV STATUS	HPV status concurrent to image collection	• -1= Unsat/No Result • 0 = HPV Negative • 1 = Known low risk HPV positive, no HPV 16 or other high risk HPV known • 2 = Known high risk HPV positive, no HPV 16 known • 3 = Known HPV16 positive

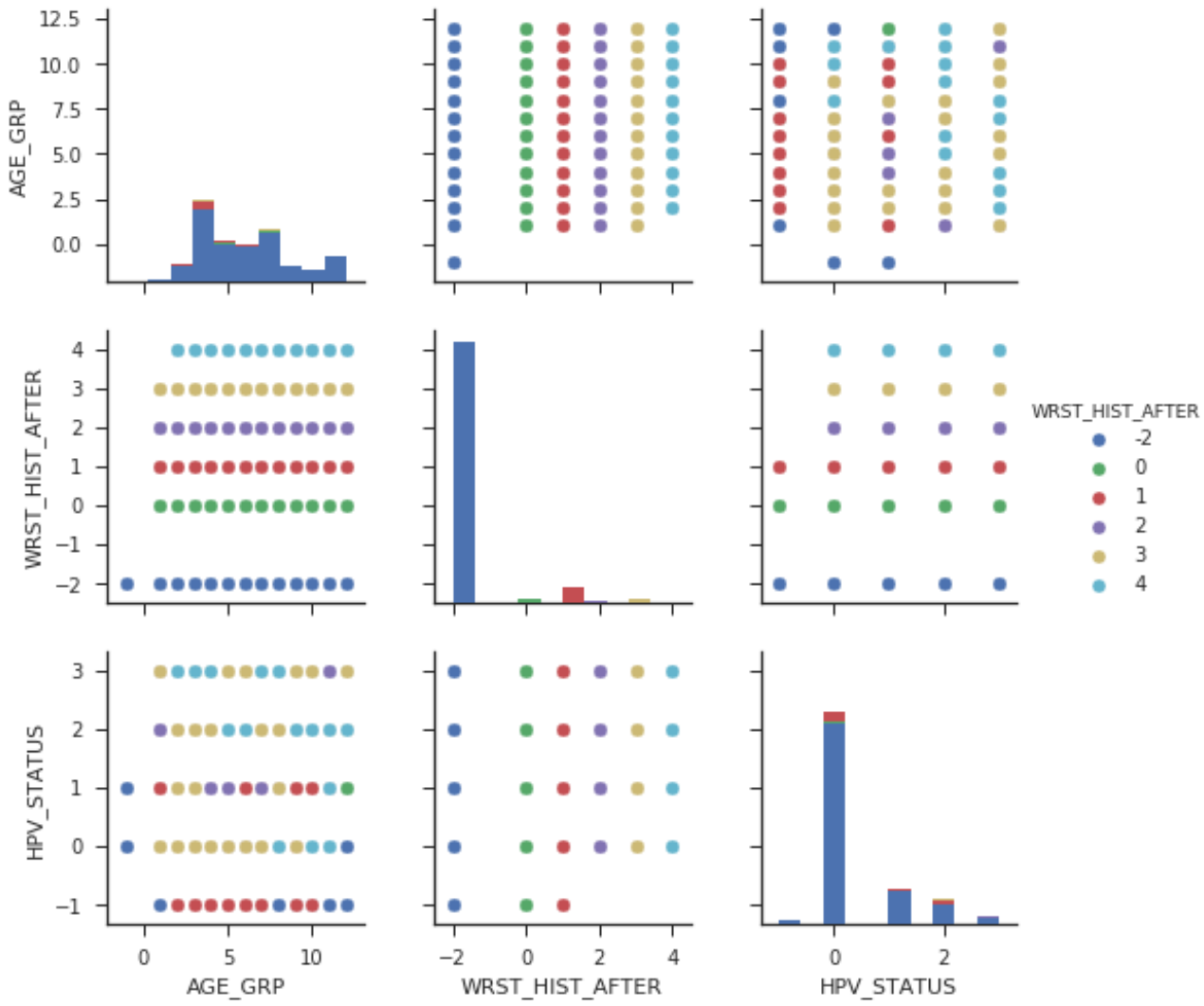

Figure 5.1: Comparisons of the fields in Table 5.1 with respect to the worst histology analysis. Each color represents different histology value. Each graph is a comparison between two fields in the data as illustrated in the x and y coordinates. The main diagonal counts the number of occurances of each data field.

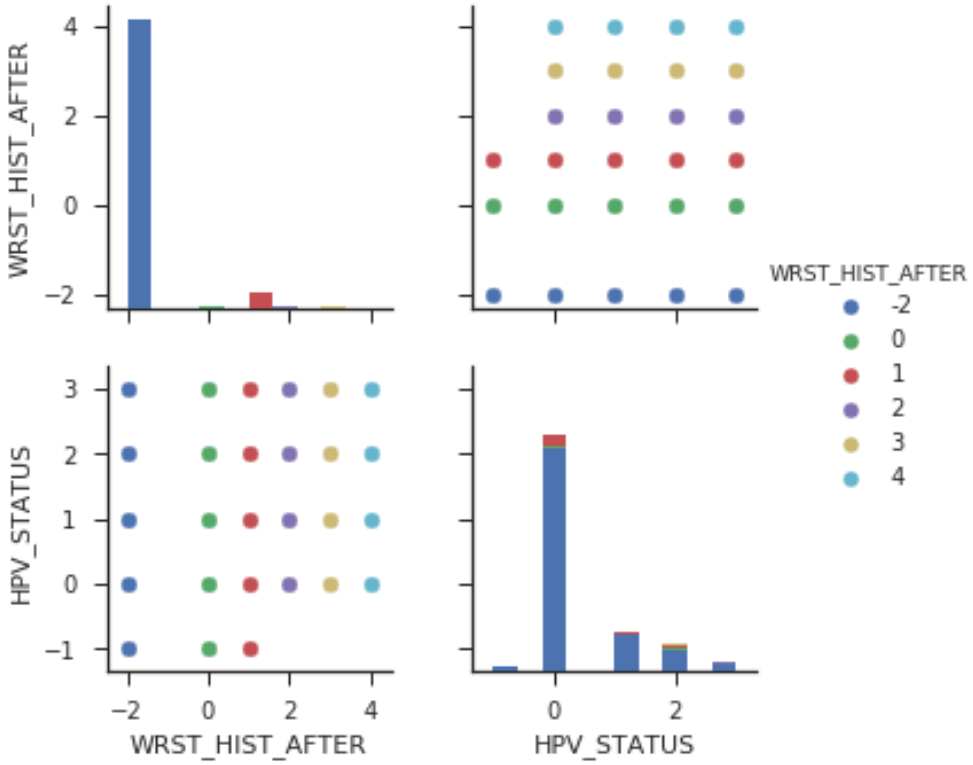

Figure 5.2: Comparing HPV and worst histology. The main diagonal represents the number of occurrences of each data field for each respective value. The other graphs compare between the histology value and the HPV status.

5.2.2 Data Collection

Since, our task is to classify cervigrams according to the CIN grade; we used the data extracted from the histology test Table 5.2 shows the possible values of the histology test.

Table 5.2: Worst Histology.

Label	**Type**
-2	No histolgoy
0	Normal
1	CIN1
2	CIN2
3	CIN3
4	Cancer

For each cervigram we associate it with one of two classes. Class 0 indicates a negative sample where there is no risk of cancer. On the other hand, class 1 indicates a risk of cancer. We separated the data according to the following piece-wise function

$$class = \begin{cases} 0 & 0 \leq \text{Label} \leq 1 \\ 1 & \text{Label} > 1 \end{cases}$$

Most of the dataset is not labeled so we only extracted the cerivgrams that have labels. Figure 5.3 shows some samples from the extracted dataset.

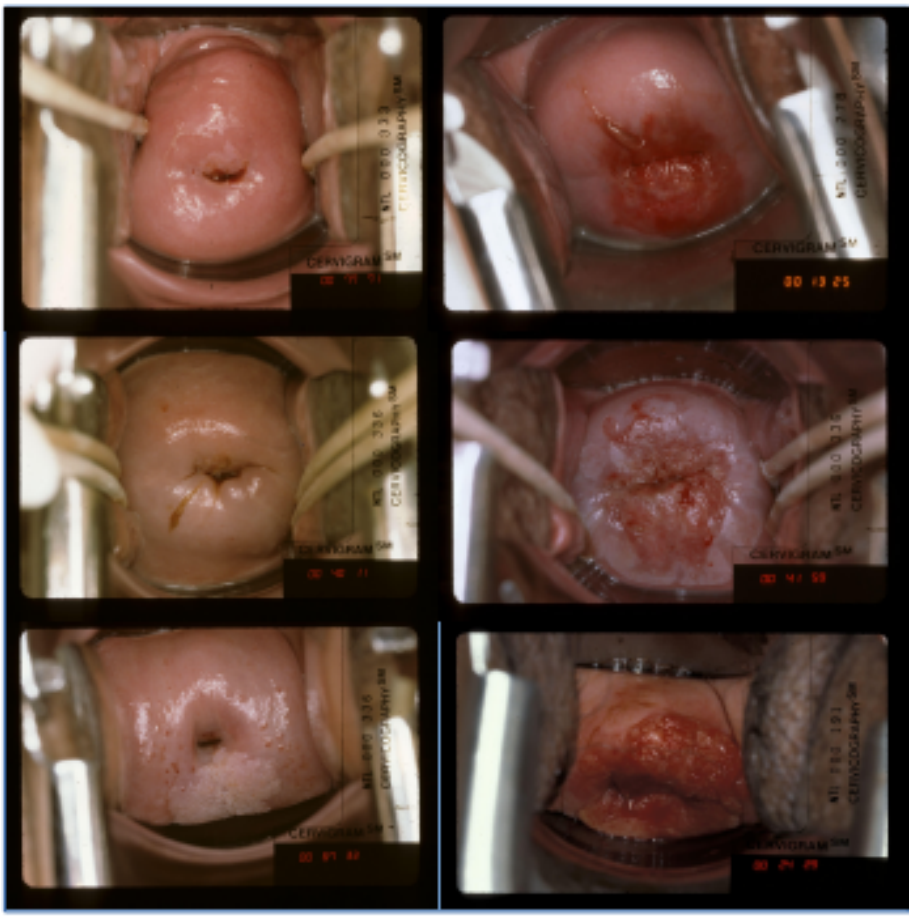

Figure 5.3: On the left we have some negative samples and on the right column we have some positive samples.

We see from the samples that we can almost indicate whether a cervigram is positive or negative but it is not that simple especially for classes CIN1, CIN2 where there seems to be a correlation between them to some extent.

5.2.3 Data Distribution

Figure 5.4 shows the distribution of the labels for the histology field if the data is available i.e it has a positive value for the filed. Notice that each patient in the dataset might have multiple images.

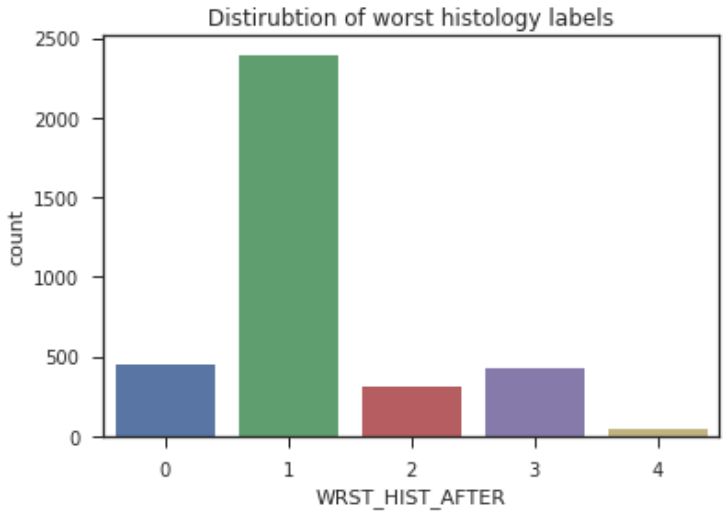

Figure 5.4: Distribution of the labels for the histology field.

We realize that most of the images are CIN1 followed by normal, followed by CIN3, CIN2 and cancer respectively. Since we care about data leaks from the training set to the validation set we had to only accept one image from each patient in the dataset. The distribution of the labels after that condition is illustrated in Figure 5.5. We made sure that the data classes are balanced as well. Table 5.3 shows the number of the extracted cervigrams per each class. All the images have dimensions 2891×1973 pixels in jpeg format. The size of each image is around 500 KB.

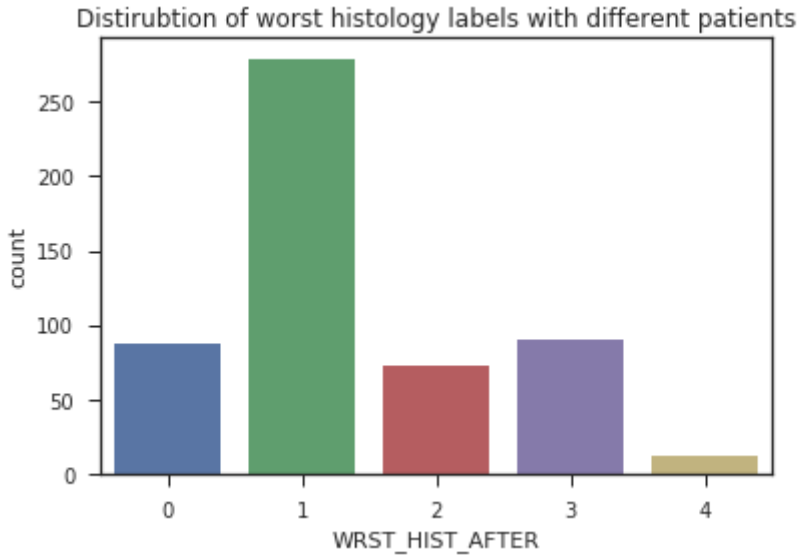

Figure 5.5: Distribution of the labels for the histology field with a simple image per patient. Since each patient has multiple images we only extract one image for each patient to avoid data leaks in the validation set.

Table 5.3: Extracted balanced dataset.

Phase / Class	**Normal/CIN 1**	**CIN2/3+**
Number of extracted cervigrams	174	174

5.2.4 t-SNE Clustering

In this section, we cluster the data using t-distributed stochastic neighbor embedding (t-SNE) [52]. t-SNE is a dimensionality reduction approach where we are interested in reducing the data from arbitrary dimensions $\mathbb{R}^n$ to lower dimensions $\mathbb{R}^2$ or $\mathbb{R}^3$ for better visualizations of the data. t-SNE is a non-linear technique that uses the probability of similarity of the pints in the high dimensional space. The algorithm calculates the conditional probability of a point A would choose another point B as its neighbor. The algorithm, attractively tries to minimize the difference of conditional probabilities. Typically, for that task it minimizes Kullback-Leiber divergence (KL-divergence) of the data points using gradient descent methods. The (KL-divergence) is a measure of the divergence rate of different probability distributions. In this section we use t-SNE to cluster different features extracted from the images in the dataset.

Using RGB Featuers

For RGB images we first flatten the images by changing the shape to $[N, r * g * b]$ where N is the number of the images and r, g, b are the image channels respectively. t-SNE is then used with the default parameters from the sklearn library to reduce the dimensionality to $\mathbb{R}^2$ which is a kind of a projection technique from high dimension to a lower dimension as we described in the previous paragraph. Given a feature vector

$\mathbb{R}^k$ we use t-SNE to reduce the dimension i.e

$$tSNE(\mathbb{R}^k) = (x, y)$$

We call x the first component and y as the second components. These components don't have a physical meaning and we cannot know what they capture. They are just used for visualization using distance. In Figures 5.6, 5.7, 5.8 and 5.9, we compare the the color channels. The labels: positive or negative represents the label of each image and they are written in the centeroid of each cluster.

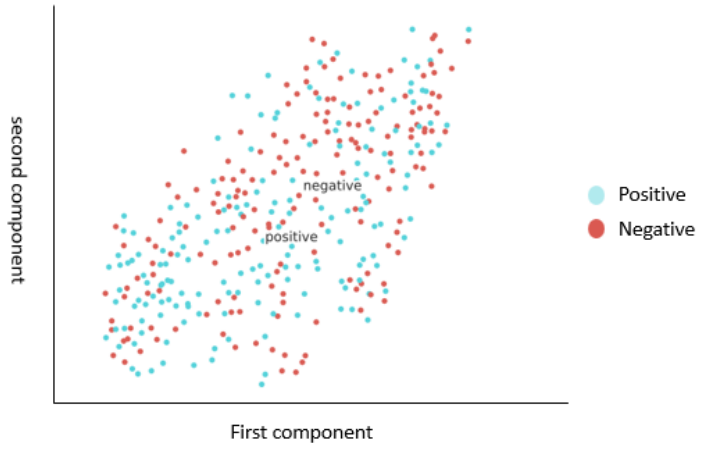

Figure 5.6: t-SNE distirubtion for the first color channel. The x coordinate represents the first component in the projected space and the y axis is the second component in the projected space.

From the graphs we conclude that the color features in the RGB space can't be good features for better understanding the data. We see that the data points are uniformly distributed across the two clusters so there is no significant difference between the two different labels.

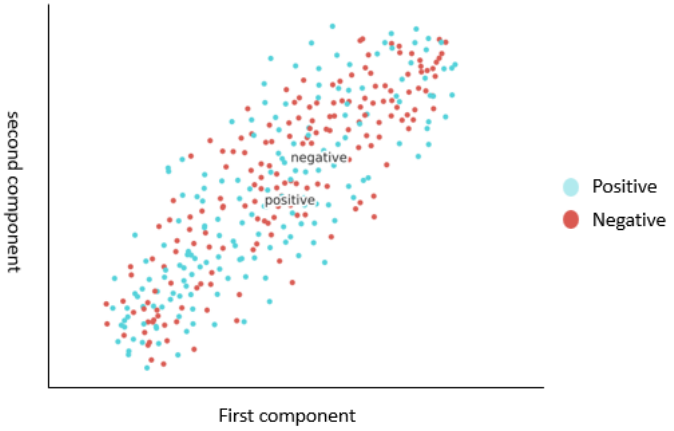

Figure 5.7: t-SNE distribution for the second color channel.

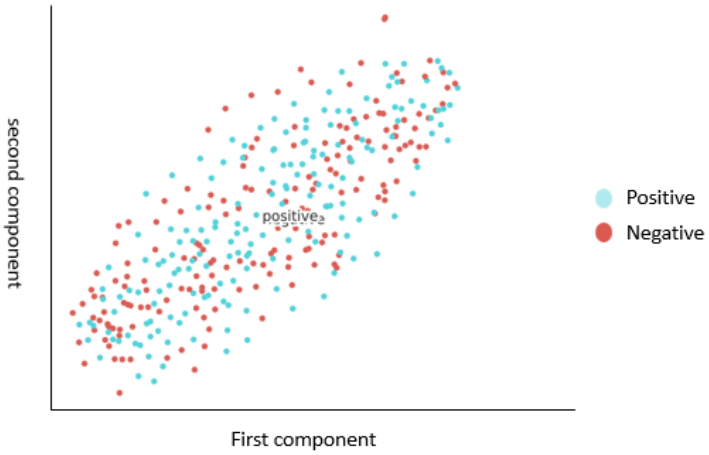

Figure 5.8: t-SNE distribution for the third color channel.

t-SNE for PHOG, PLAB, PLBP Features

We compared between the hand crafted features: locally binary patterns (PLBP), pyramid histogram in the L*a*b* color space (PLAB) and pyramid histogram of oriented gradients (PHOG). In this section we compare the hand crafted features by looking at the clusters generated by t-SNE.

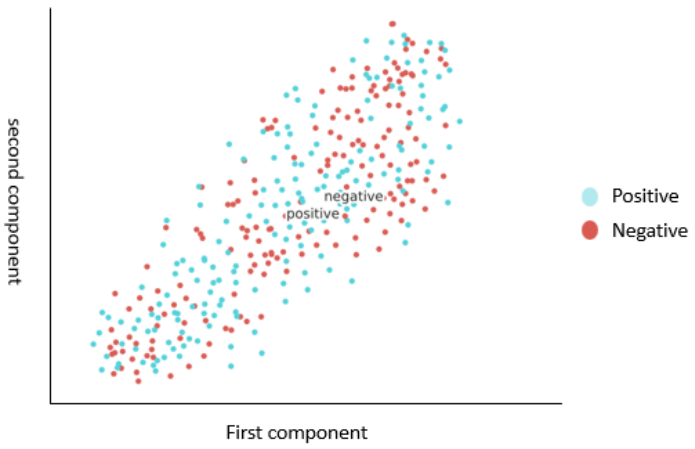

Figure 5.9: t-SNE distribution for all RGB color channels.

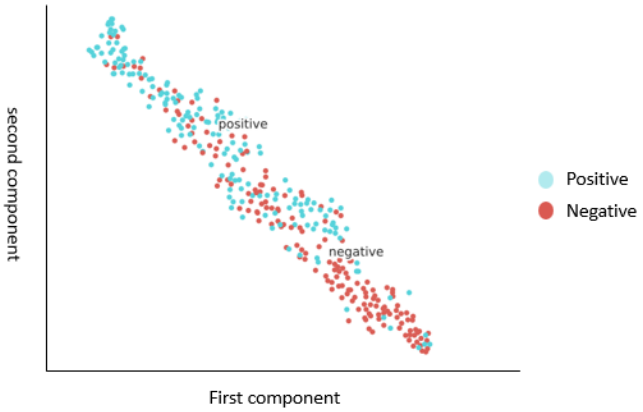

Figure 5.10: t-SNE for PLBP features.

We conclude that PLAB+PLBP seem to be better descriptors than PHOG. As we can see from the graph of PLAB the labels are grouped into different clusters. This is less clear in the PLBP descriptor but it seems there is a boundary between the classes with some outliers. On the other hand, PHOG descriptor does not seem to differentiate between the two classes. However, this is only in the $\mathbb{R}^{\not\vdash}$ space but it might differentiate in higher dimensional spaces. These are initial guesses and can not

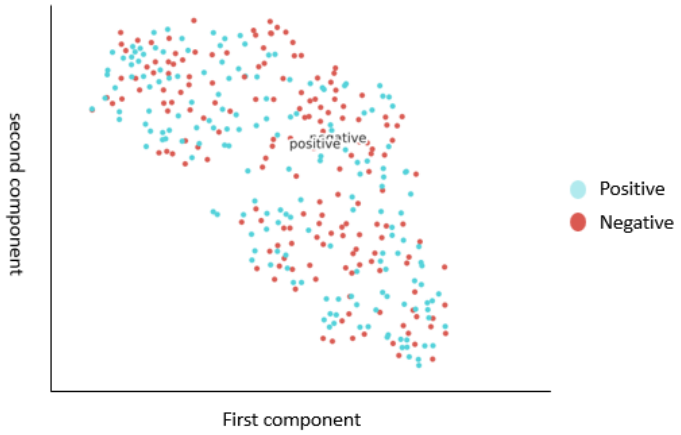

Figure 5.11: t-SNE for PHOG features.

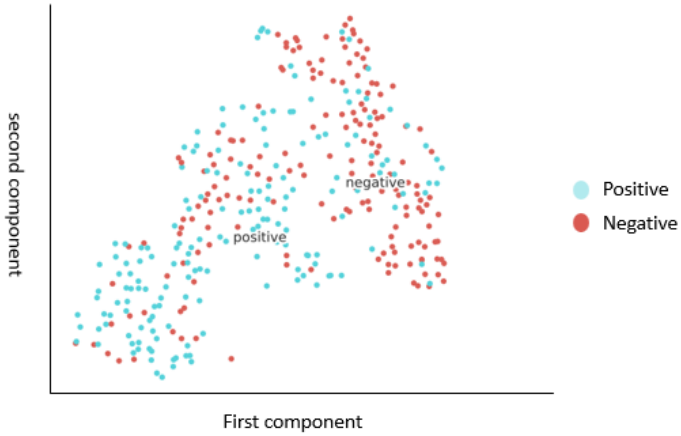

Figure 5.12: t-SNE for PLAB features.

be used to give a clear idea if PLAB and PLBP are better than PHOG but they are certainly a confirmation that the first two descriptors are good features.

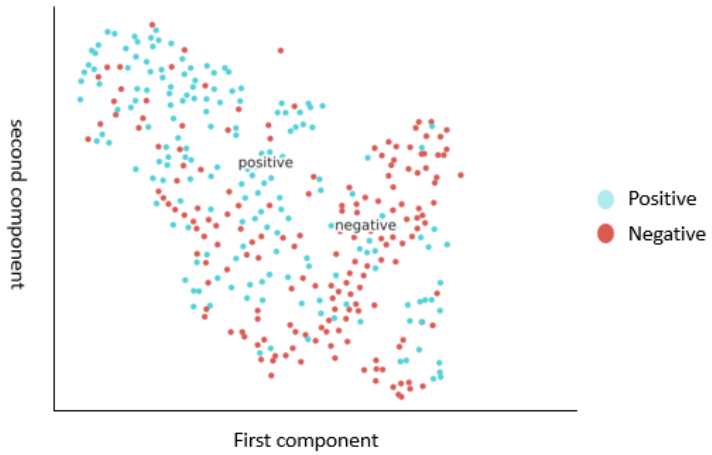

Figure 5.13: t-SNE for PLAB+PHOG+PLBP features.

5.2.5 Data Augmentation

Image augmentation refers to the process of increasing the ability of the model by adding some transformations of the training dataset. In other words, we add some perturbation to the image by some transformation such that we don't change the true class of the image [47]. Data augmentation has been popular in deep convolutional neural network since their usage in ImageNet challenge. In AlexNet, they used translations and horizontal reflections which increased the number of training samples up to a factor of 2048 [34]. Another form of augmentation was also used in forms of PCA on the group of RGB set in the training data [34]. In [41] the authors described using some other forms of data augmentation. They used a random crop of certain size or its horizontal flip and added to the training set [41]. More complex models also exist to generate data from training set. GANs or generative adversarial networks provide an approach to generate realistic images out of a certain data set [53]. The main idea is creating two neural networks one called the discriminator which classifies an image as real or fake and the other network called the generator which generates fake images from noisy data and tries to fool the discriminator to classify them as real [53].

In our approach we use different types of augmentation approaches like random cropping, random flipping and random rotation. These approaches are illustrated in Figure 5.14. We noticed a small jump in the accuracy of the results when we used augmentation to the dataset. This is due to the fact that CNNs require a large number of training data and since we have a small number of cervigrams the

usual transformation approaches help in gather different features by applying different transformations.

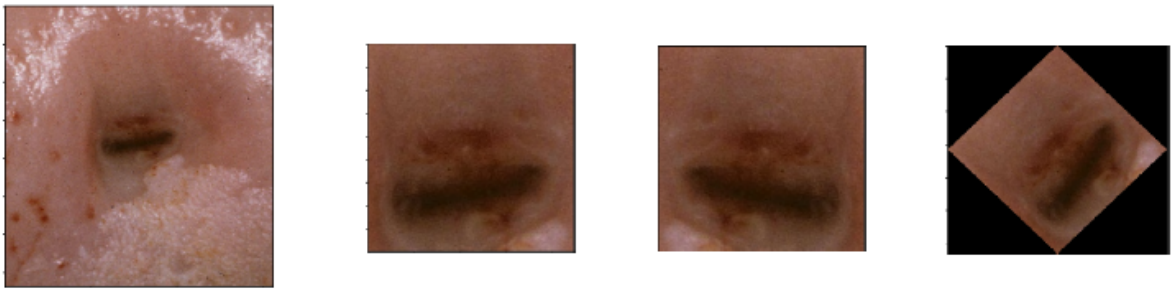

Figure 5.14: 256 x 256 random cropping followed by horizontal flipping followed by 45 degree rotation. Since we are applying random cropping some features might not exist in the current augmented batch but will exist in other batches in different epochs.

5.3 Results and Discussion

5.3.1 Segmentation

Here we compare our method to other methods in the literature and show that our method is much faster than the other methods. We try to compute the average time taken to predict the resulting bounding box for a certain method. Basically, the methods in the literature are data-driven and don't use NN for object detection. Song et al. used 939 labeled images as a database for labeling a new image [25]. They calculate the PHOG descriptor of the test image and all the other images in the database. Then, they find $k = 20$ matching images using a similarity measure. Then calculate the bounding box as the average of the k bounding boxes. Kim and Huang suggest a more sophisticated approach [24]. For each image in the database they calculate color and texture feature descriptors. For each test image they compare its descriptors against all the images in the database and choose the M matching images. For each matching image the descriptors are recalculated inside the bounding box and compared with the corresponding bounding box in the test image. The bounding box with the highest score is chosen as the bounding box of the test image. They compared this method to other variants like Average Bounding Box and Image Bounding Box methods. Image Bounding box methods finds the image in the labeled dataset with the highest similarity score. Average bounding Box method uses typically the same approach as in [25].

In table 5.5 we compare our method to other methods in the literature in terms of intersection over union (IoU) and time of inference. We see that our method is much faster using YOLO. Moreover, our method's inference time is independent of the number of the images in the dataset with bounding box labels because at prediction time we already have the trained model. On the other hand, other approaches which are highly-dependent on the number of the labeled images will take a longer time because they don't have a trained model. They will have to evaluate a similarity measure with all the labeled images. Moreover, the similarity measure adds another constraint especially when using a computationally expensive similarity measures. This computation time when increasing the size of the labeled images only matters for us when training the model which does not affect the inference time at all. Furthermore, since our method uses deep networks it is robust to variance in the dataset because of using data augmentation in the training time. However, the other approaches will suffer from overfitting to the images in the dataset. This is a serious problem especially when we have blurry, deformed, rotated or transformed test images which will cause the prediction model to fail.

Table 5.4: Comparing IoU and prediction time against other methods in the literature.

Method	**IoU**	**Average time** (*minutes*)
Image Bounding Box [54]	0.611	4
Optimized Bounding Box [24]	0.736	20
Average Bounding Box [25]	0.699	4
Ours with YOLO	0.68	0.00367

Our approach causes the model to have lower IoU compared to other methods which is understandable since we used less data compared to the other approaches. In the approach by Kim and Huang they used 25 % more data than what we used which caused they more model to have a larger IoU. We have to pay the price of faster prediction by having less IoU measure. We believe that IoU could be improved drastically by having a larger dataset. Moreover, we think that the confidence score reported during testing could be used as a good measure for practitioners to know if the model is having any problems for detecting a good bounding box predictions for the current test image. Figure 5.15 shows some boxes with their confidence scores in percentages. We realize that the YOLO model can easily detect the opening of the cervix. Sometimes it gets confused when there are some blood or deformed tissues. Note that the confidence score is evaluated using the formula

$$\Pr(\text{Object}) \times \text{IoU}_{\text{pred}}^{\text{truth}}$$

where Pr is the probability of the object and IoU is the intersection over union metric. The probability of the object is calculated as a value in the interval $(0, 1)$ as an output of the YOLO Network. Also, our method might detect more than bounding boxes as we see in some images in Figure 5.15. We usually take the bounding boxes with the highest confidence score. For instance the in the image with two bounding boxes one with 50% and the other one with 70% we choose the latter bounding box as our predicted bounding box to the current image.

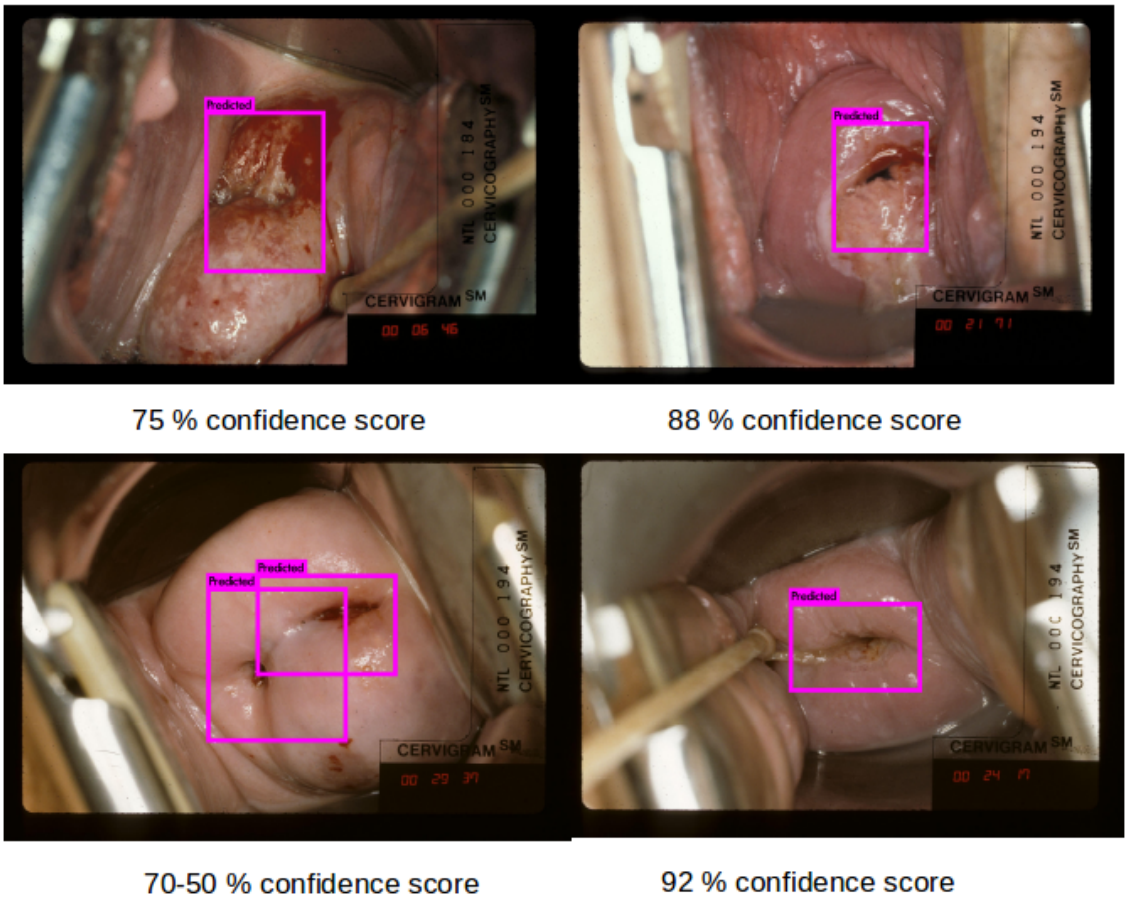

Figure 5.15: The confidence scores of some cervigrams bounding boxes using YOLO.

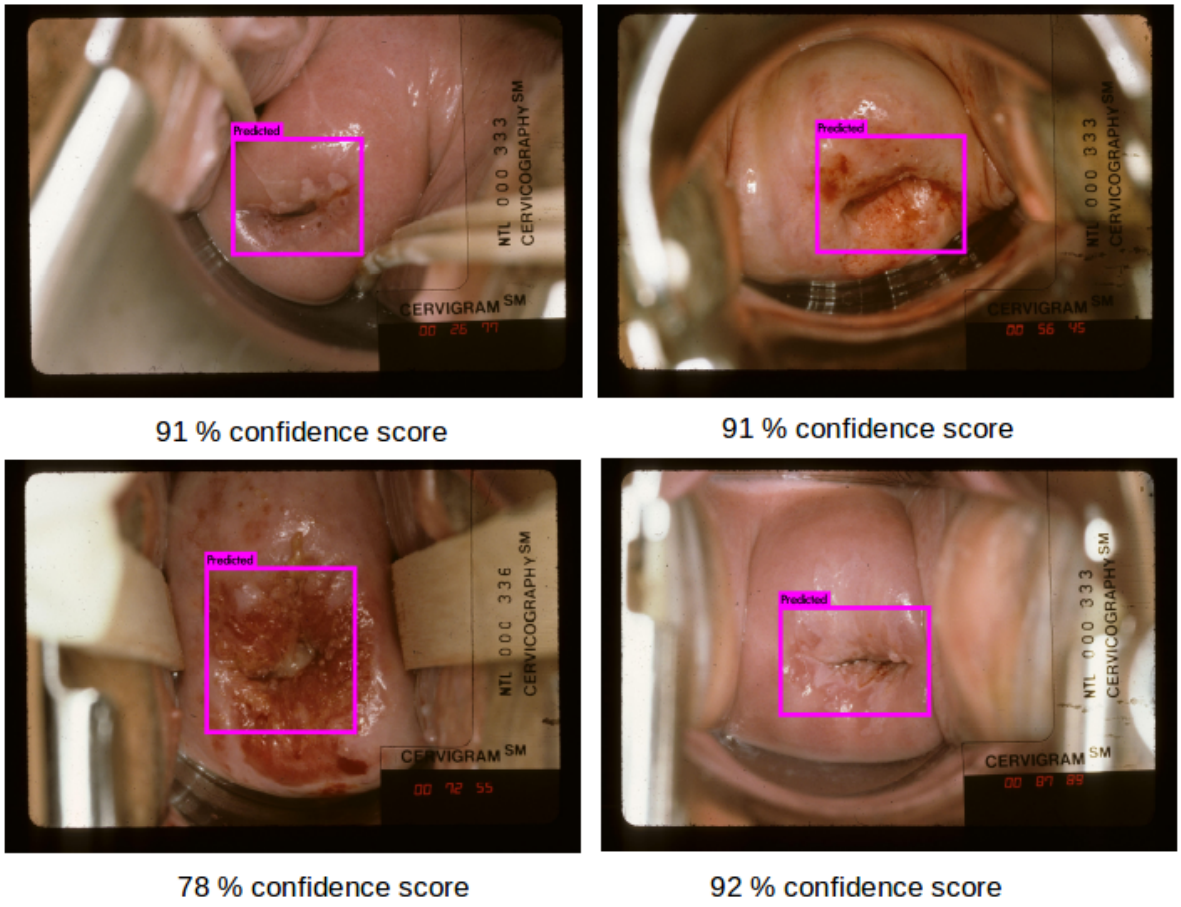

Figure 5.16: More confidence scores samples.

5.3.2 Hand Crafted Features

The hand-crafted features are extracted using PLAB, PHOG and PLBP described in **section 2.3.3**. After segmenting the RoI we resize the image to size (300, 250). Then we extract the features for the image as the following:

- PLAB, calculates a pyramid of histograms in the L*A*B* color space. Three pyramids are extracted by dividing the images into three different levels. The levels include 1, 4, 16 sub-regions respectively at each level. For each level a histogram of 16 bins is calculated for each channel in the L*A*B* space. This results in a descriptor of size $(16 + 4 \times 16 + 16 \times 16) \times 3 = 1008$.

- PLBP, we used $P = 8, R = 1$. The descriptor was evaluated on the gray scale image by dividing the image into regions of sizes 1, 4, 16 and 64. The number of bins is 10 at each region. This resulted in a descriptor of size $10 + 4 \times 10 + 16 \times 10 + 64 \times 10 = 850$.

- PHOG, The last descriptor is a PHOG which evaluates a pyramid histogram of oriented gradients. A binary edge image is first calculated using sobel edge detector. The levels of the pyramid are 1, 4, 16 and 64. The number of the bins is 8. This results in a descriptor of size $8 + 4 \times 8 + 16 \times 8 + 64 \times 8 = 680$.

Hence, by concatenating the three descriptors we obtain a vector of size $1008 + 850 + 680 = 2538$. We then feed the vector two a neural network. The neural network

has an input shape equals to 2538 and and output shape equals to 1. We used different number of hidden layers.

- **Model 1**, this model contains one hidden layer with 128 units followed by a ReLU activation function. Then a dropout layer with rate 0.5 is attached. The outputs are contains only one neuron followed by a sigmoid activation.
- **Model 2**, in this model we have two hidden units with the number of units 128 and 256 respectively. There is a dropout layer in-between them with rate 0.5. Each hidden unit is followed by a ReLU activation. The output layer contains only one neuron followed by a sigmoid activation.

We used an adam optimizer with binary cross entropy loss function defined as

$$-\sum_{i} y_i \log(p_i)$$

Where y_i takes the value 0 or 1 and p_i is the outputs of the sigmoid activation.

5.3.3 Automatic Feature Extraction

We used two types for convolutional neural networks

- **Model 3**, with 2 convolutional layers each with filters of size (3,3) and strides of size (2,2) and the number of filters is 16 and 32 respectively. The conv layers are followed by ReLU activations and max-pooling layers which reduce the ssize into half. The output of the conv layers is flattend into a dense layer with 128 units followed by a tanh activation and a dropout layer with rate 0.5. The final layer contains one unit followed by a sigmoid activation.

- **Model 4**, with 3 convolutional layers each with filters of size (3,3) and strides of size (2,2) and the number of filters is 16, 32 and 64 respectively. The conv layers are followed by ReLU activations and max-pooling layers which reduce the size into half. The output of the conv layers is flattend into a dense layer with 128 units followed by a tanh activation and a dropout layer with rate 0.5. The final layer contains one unit followed by a sigmoid activation.

We used an adam optimizer with binary cross entropy loss function as in the hand crafted features.

5.3.4 Training

Table 5.5 shows the results of training all of the four models. We report the accuracy, specificity and sensitivity at a 0.5 threshold. These results are averaged over each

cross-validation step and the standard deviation is also calculated.

Table 5.5: Model 1: hand crafted features with 1 hidden layer, Model 2: hand crafted features with 2 hidden layers, Model 3: CNN with 2 conv layers and 1 hidden layer, Model 4: CNN with 3 conv layers and 2 hidden layers

Model	**Accuracy**	**Specificity**	**Sensitivity**
Model 1	72.94 ± 6.94	76.80 ± 5.95	68.96 ± 9.96
Model 2	77.06 ± 7.06	77.97 ± 5.22	75.22 ± 11.28
Model 3	68.24 ± 9.74	77.43 ± 10.57	59.70 ± 12.08
Model 4	70.29 ± 8.57	68.33 ± 14.09	72.30 ± 13.85

We also report the Receiver operating characteristic (ROC) graph in Figure 5.17. We plot TPR against 1 - TPR for all the models for different thresholds in the binary classifier. The threshold defines a probability distribution where values below the threshold are chosen as negative and values above the threshold are chosen as positive. A binary classifier classifies input data as positive or negative by comparing the output value to the threshold. We see that **model 2** with two hidden layers achieves the best value of the AUC metric and across all other metrics. However, the evaluation of the descriptors is a very slow process as it takes around 1.3 seconds for each image to extract PLAB+PHOG+PLBP features while it takes 0.008 second to evaluate the CNN features which is 160 times faster.

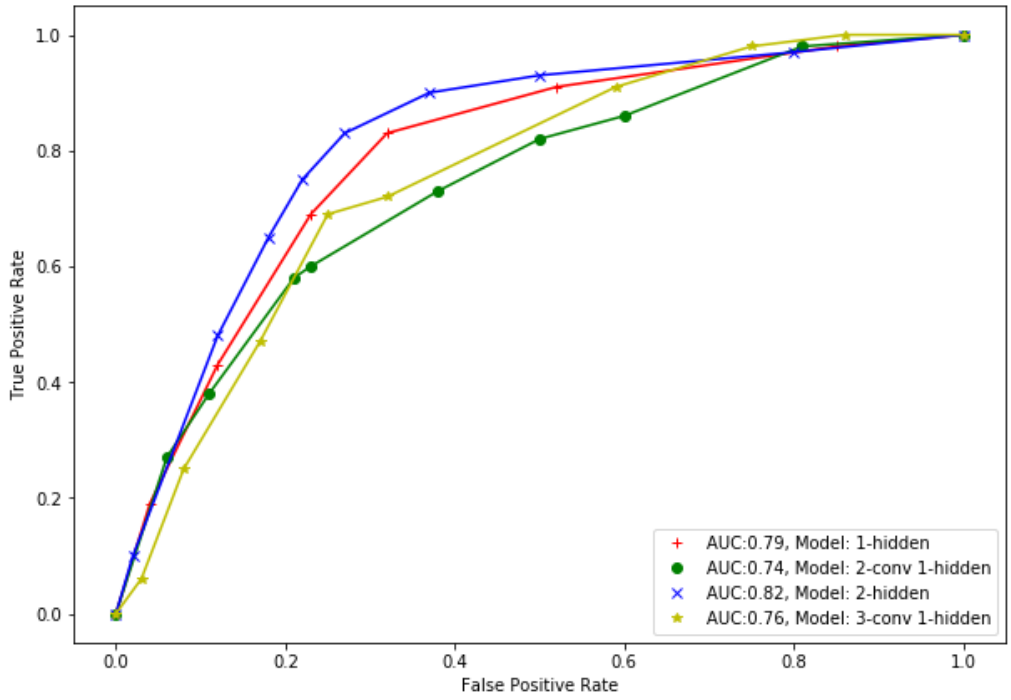

Figure 5.17: ROC graph with AUC values for each model.

5.3.5 Failed Cases

Segmentation

Figure 5.18 shows the distribution of the confidence values for segmentation across all the images in the dataset. We realize that most of the images have higher confidence interval > 80 %.

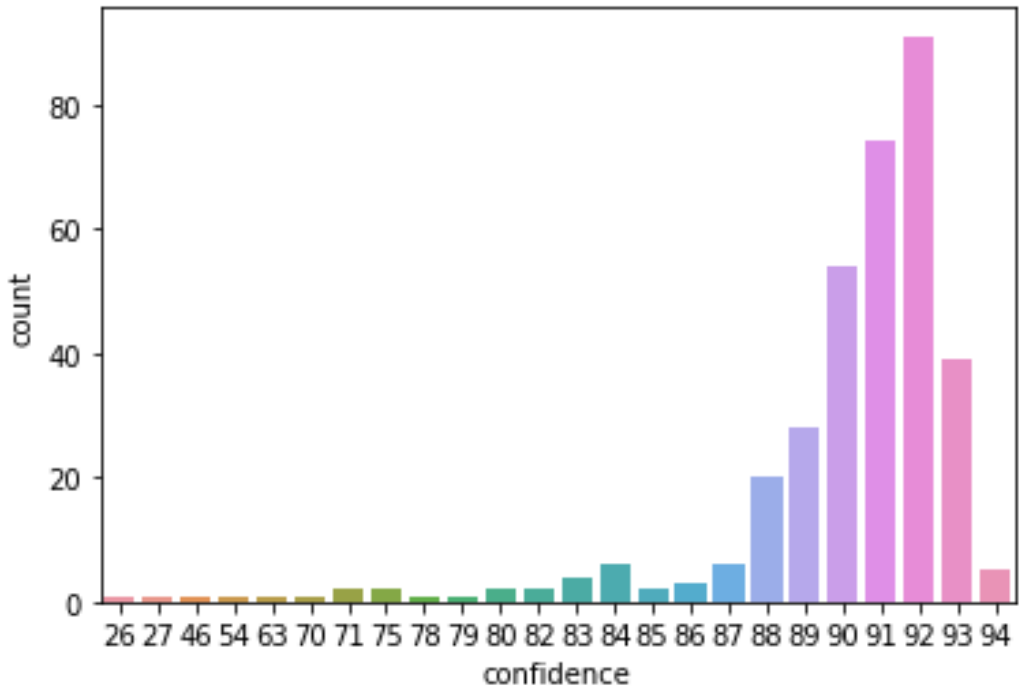

Figure 5.18: Confidence values for all the data.

In Figure 5.19 we show some bounding box predictions for images with confidence values less than 80 %. We believe that these images need further inspection in order to correct the bounding boxes.

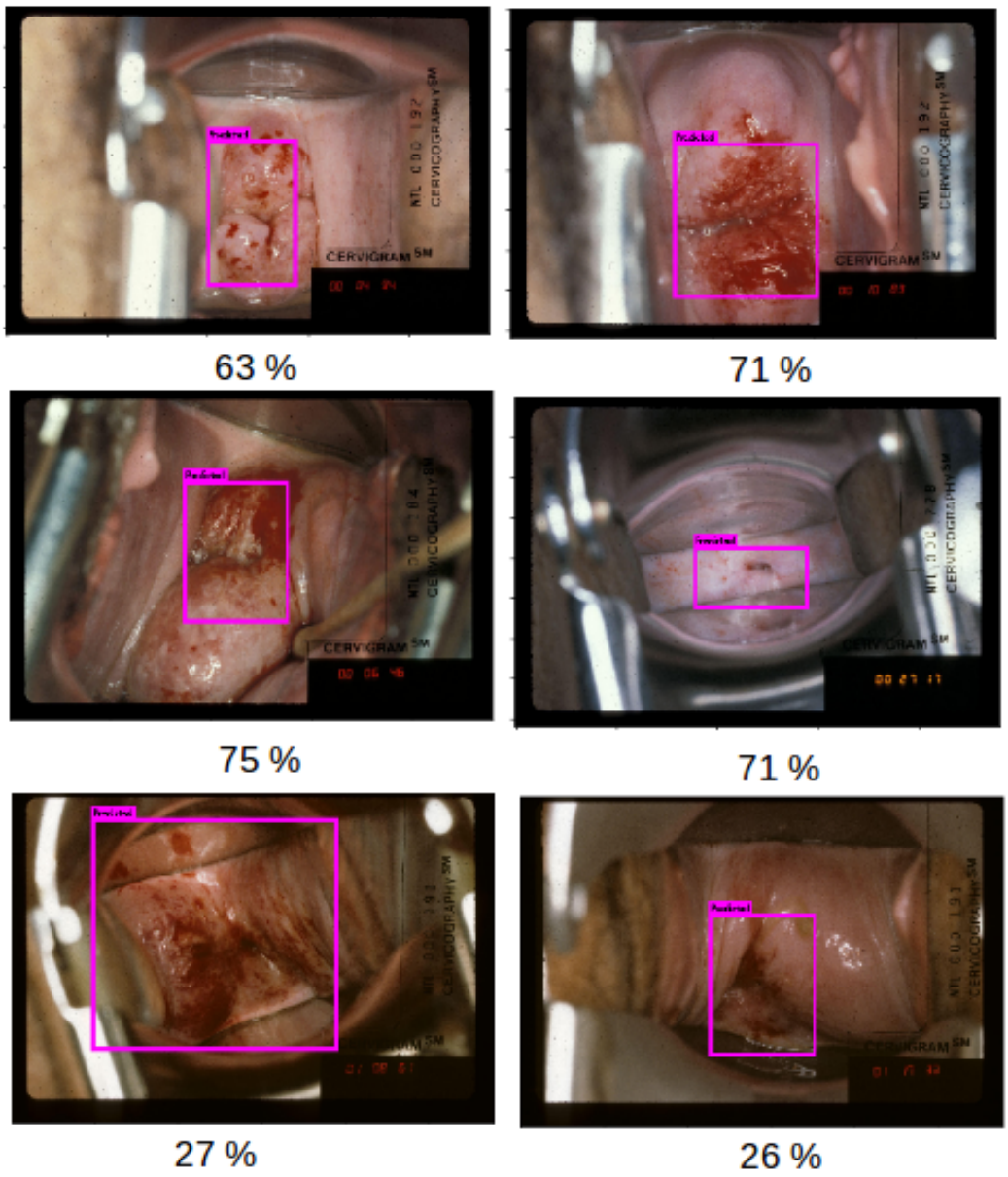

Figure 5.19: Some bounding box predictions with less than 80 % confidence.

Classification

False positive images are images which are predicted by the model to be positive but have a negative truth label. On the other hand, False negative images are those which are predicted as negative but have a positive truth label. We know that the classifier of our model predicts a probability distribution according to the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Hence the features will be fed into the binary classifer and result into x which is fed into the sigmoid function. This results in a probability distribution $p(c_0|x) = p$ and $p(c_1|x) = 1 - p$ where c_0 refers to the negative class and c_1 refers to the positive class. We used $p = 0.5$ as the threshold of our model with the values of $p \leq 0.5$ are classified as negative classes and $p > 0.5$ are classified as a positive class.

Figure 5.20 shows false negative and positive samples along with their probabilities according to the predicted class. We realize that some cervigrams with a negative class have a big probability $>> 0.5$ and some cervigram with a poisitve class have a small probability $<< 0.5$. On the other hand some cervigrams are in the boundary with small epsilon ϵ and probability $0.5 - \epsilon < p < \epsilon + 0.5$.

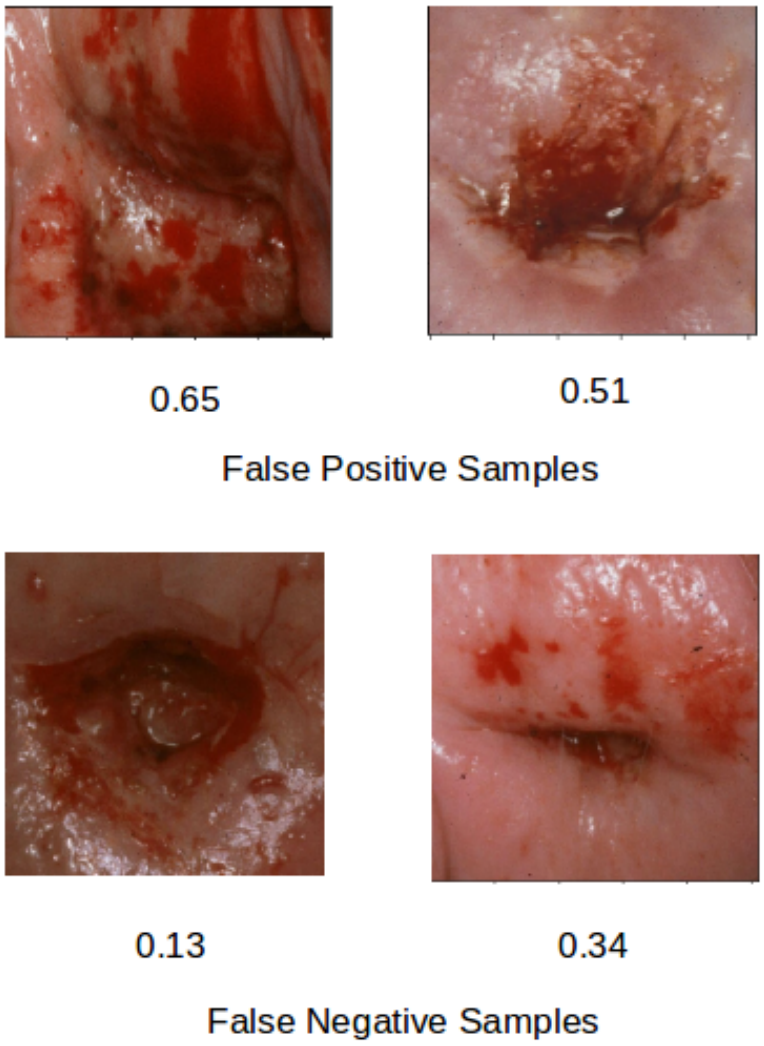

Figure 5.20: Samples of false positive and false negative cervigrams with the probability value.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

We provided an overall architecture to solve cervical cancer segmentation and classification. We suggested a fully automated approach for cervical cancer segmentation and classification using neural networks. The process summarized data collection, segmentation, preprocessing, augmentation, feature extraction and classification. First, we collected a labeled dataset that associates each cervigram with either positive or negative class. We then provided a segmentation approach for detecting the area of interest in the cervigram. Our approach provided a state of the art method in terms of speed with a comparable performance. Also, Our approach is independent of the size of the dataset we are training on. We proved that our segmentation method is 10^3 faster than the approaches available in the literature for such domain. We also compared hand-crafted features to convolutional neural networks in such domain.

6.2 Future Work

We believe that we can extend this work by working on a larger dataset. The main problem that we faced was that the number of labeled images was small which caused deep features not to perform as we expected. Also, we can prove that the models we created could be easily deployed into mobile devices. Furthermore, we could improve the speed of the segmentation approach which is basically the bottleneck of our pipeline by using a custom model with separable convolutional layers. These layers proved to be faster and contain much less parameters of the original model. In addition to that, we can compare different segmentation models like Mask-RCNN , Faster-RCNN and conclude whether pixel-wise segmentation could give better results than bounding box prediction methods!

APPENDIX A

MULTI-CLASSIFICATION

In this thesis we described how to classify cervical intraepithelial neoplasia into either CIN1 or CIN2/3+ which assumes a binary classifier. In this chapter we study the effect of further dividing the dataset into three classes, namely CIN1, CIN2 or CIN3. We use the same dataset but we ignore the either classes. We also pass the dataset by the segmentation process so the images are already segmented using YOLO. Note that we pass the same dataset we extracted to the previous classifiers. Figure A.1 shows the number of classes for each type.

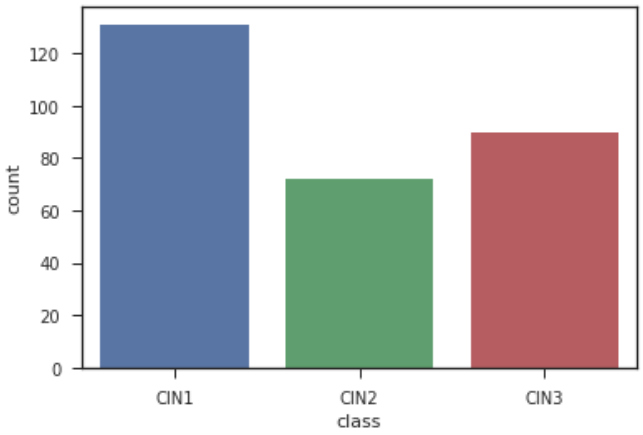

Figure A.1: Distribution of the labels for the histology extracted from our dataset.

Note that the dataset is unbalanced as we see that Nearly CIN1 is double of CIN2. We repeat our classification process as we described in the previous chapters. Note that since we have three classes the output will contain three neurons as opposed to the previous one which had only one neuron. Moreover, we used a softmax classifier which is a generalization of the sigmoid function for multi-classification. If we have K classes then the we will have a vector z with K entries. Then for each entry we convert it using the function

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{i=1}^{K} e^{z_i}} \quad j = \{1, 2, \cdots, K\}$$

Note that the vector is converted into a probability distribution where all the entries are between 0 and 1 and their sum is equal to 1. Also, we need to update our loss function into what we call categorical cross entropy defined as

$$Loss = -\sum_{x} p(x) \log(q(x))$$

where p is the true distribution of the data and q is the predicted result. Since this function requires the true labels to be a probability distribution we must convert them using one hot encoding where the mapping is

$$0 : [1, 0, 0], \ , 1 : [0, 1, 0], \ 2 : [0, 0, 1]$$

Hence the loss function will a high positive value if the predicted value is not close to 1. Note that the cross-entropy function is a smooth function is it is preferred to

be used for classification tasks. Now we are ready for classification. We use the same parameters as in the previous chapter except that we convert the last layer to have three neurons followed by a softmax activation. We keep the other parameters of the models the same. Figure A.2 illustrates that the models fails to recognize class CIN2 which is totally expected since it is very close to both classes.

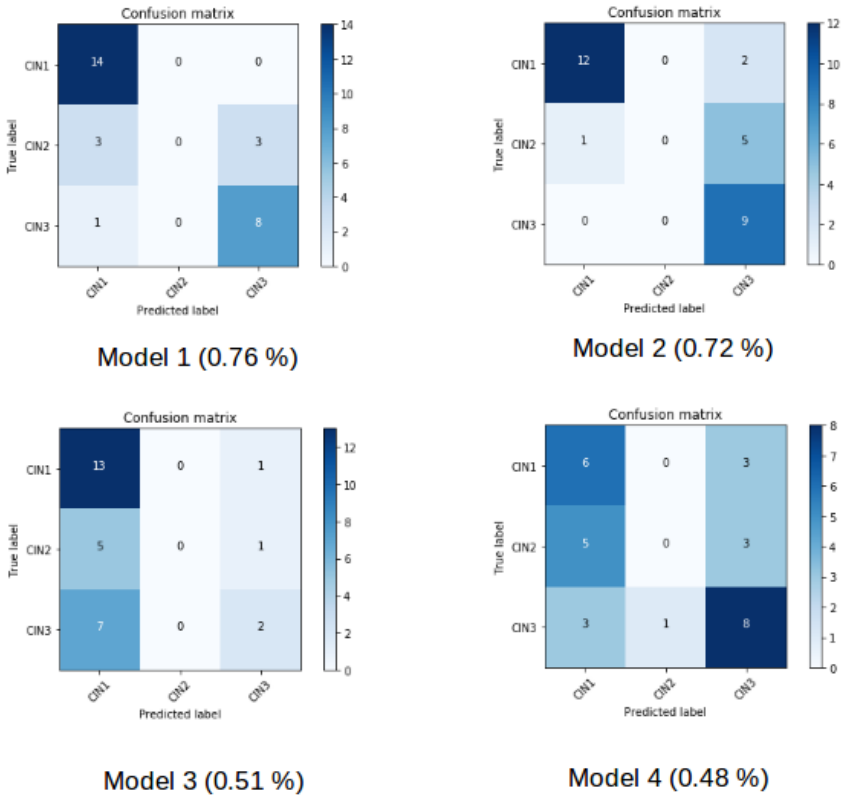

Figure A.2: Comparisons between the four models with 3 classes.

Figures A.3, A.4 show the same metrics but for different number of classes. Notice that due to the huge unbalance of the data, it seems that the model is only learning for CIN1 and CIN3 which contain the largest number of classes.

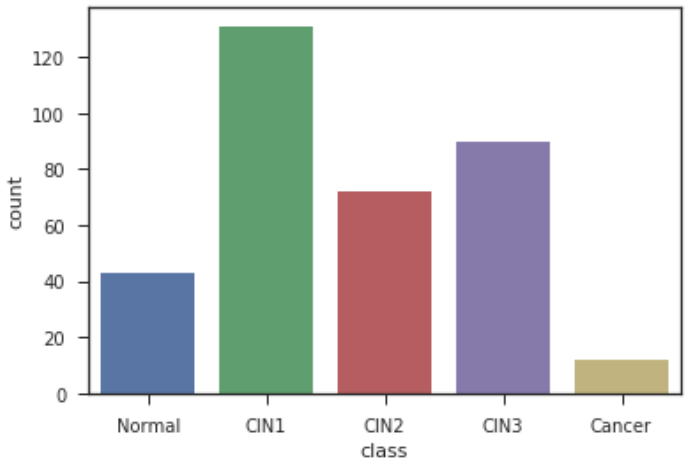

Figure A.3: Distribution of the labels for 5-way classification.

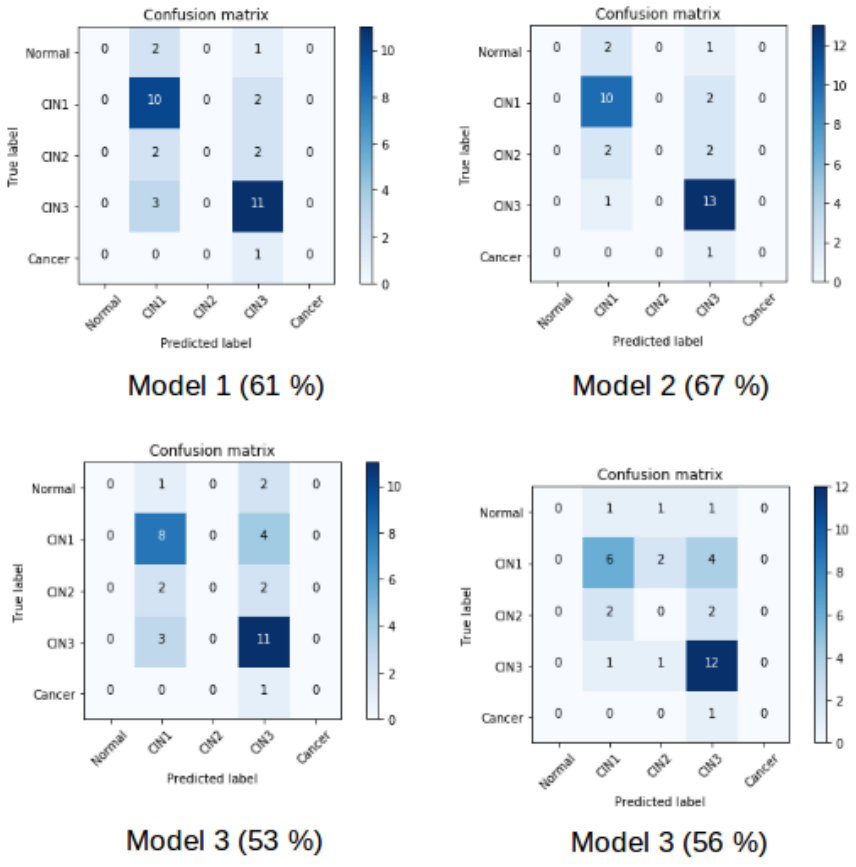

Figure A.4: Comparisons between the four models with 5 classes.

APPENDIX B

IMBALANCED CLASSIFICATION

In this section we consider the possibility of using the full dataset for classification. Figure B.1 shows the distribution of the labels for the full dataset. Notice that CIN2 contains most of the cervigrams hence there is a bias for predictions towards this class. Figures B.2 and B.3 show the results of the classifications considering binary and 5 way classification.

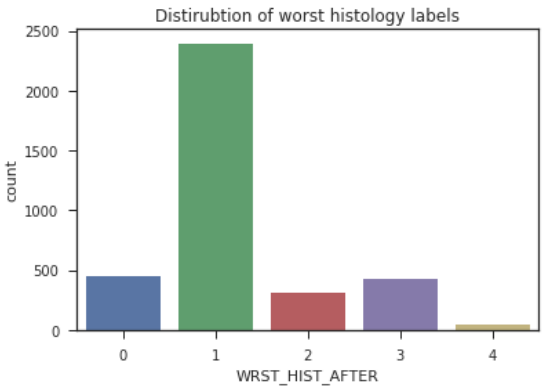

Figure B.1: Distribution of the labels for the histology field for the full dataset.

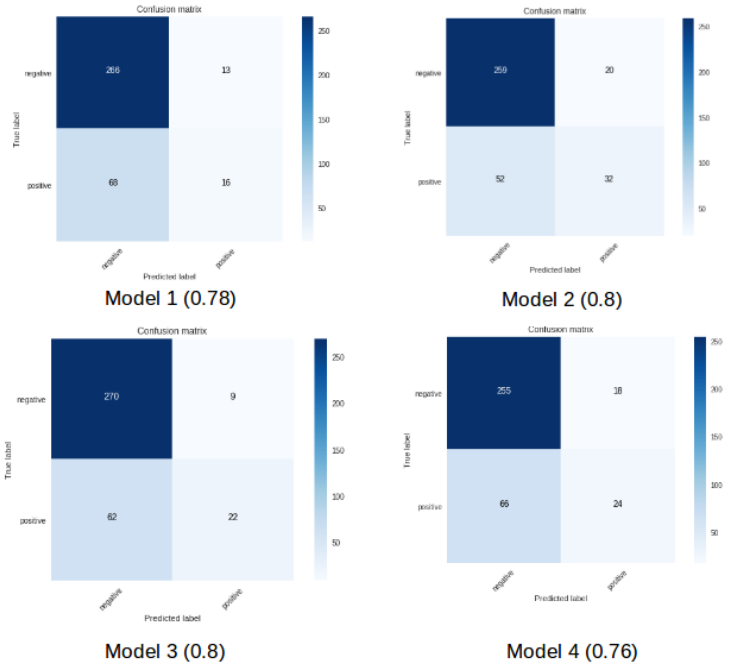

Figure B.2: Comparisons between the four models with two classes.

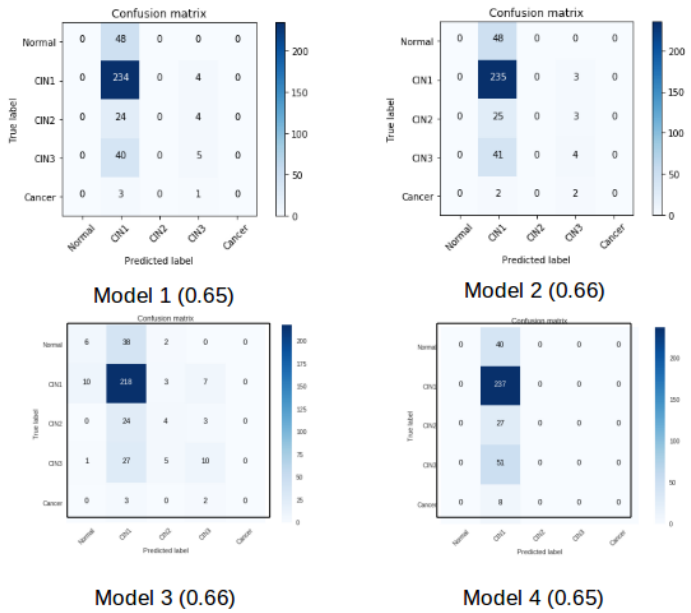

Figure B.3: Comparisons between the four models with five classes.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, vol. 4, p. 12, 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2980–2988, IEEE, 2017.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[9] L. A. Torre, R. L. Siegel, E. M. Ward, and A. Jemal, "Global cancer incidence and mortality rates and trends—an update," *Cancer Epidemiology and Prevention Biomarkers*, vol. 25, no. 1, pp. 16–27, 2016.

[10] A. W. LaVigne, S. A. Triedman, T. C. Randall, E. L. Trimble, and A. N. Viswanathan, "Cervical cancer in low and middle income countries: Addressing barriers to radiotherapy delivery," *Gynecologic oncology reports*, vol. 22, pp. 16–20, 2017.

[11] S. Wittet, S. Goltz, and A. Cody, "Progress in cervical cancer prevention: the cca report card 2015," *Seattle, WA: Cervical Cancer Action*, 2015.

[12] M.-H. Mayrand, E. Duarte-Franco, I. Rodrigues, S. D. Walter, J. Hanley, A. Ferenczy, S. Ratnam, F. Coutlée, and E. L. Franco, "Human papillomavirus dna versus papanicolaou screening tests for cervical cancer," *New England Journal of Medicine*, vol. 357, no. 16, pp. 1579–1588, 2007.

[13] L. G. Koss, "The papanicolaou test for cervical cancer detection: a triumph and a tragedy," *Jama*, vol. 261, no. 5, pp. 737–743, 1989.

[14] K. K. Vesco, E. P. Whitlock, M. Eder, J. Lin, B. Burda, C. A. Senger, R. S. Holmes, R. Fu, and S. Zuber, "Screening for cervical cancer: a systematic evidence review for the us preventive services task force," *Lancet Oncol*, vol. 12, no. 7, pp. 663–72, 2011.

[15] J. Jordan, A. Singer, H. Jones, and M. Shafi, *The cervix*. John Wiley & Sons, 2009.

[16] S. De Sanjosé, B. Serrano, X. Castellsagué, M. Brotons, J. Muñoz, L. Bruni, and F. Bosch, "Human papillomavirus (hpv) and related cancers in the global alliance for vaccines and immunization (gavi) countries. a who/ico hpv information centre report," *Vaccine*, vol. 30, no. Suppl 4, pp. D1–83, 2012.

[17] F. X. Bosch, M. M. Manos, N. Muñoz, M. Sherman, A. M. Jansen, J. Peto, M. H. Schiffman, V. Moreno, R. Kurman, and K. V. Shan, "Prevalence of human papillomavirus in cervical cancer: a worldwide perspective," *JNCI: Journal of the National Cancer Institute*, vol. 87, no. 11, pp. 796–802, 1995.

[18] J. M. Walboomers, M. V. Jacobs, M. M. Manos, F. X. Bosch, J. A. Kummer, K. V. Shah, P. J. Snijders, J. Peto, C. J. Meijer, and N. Muñoz, "Human papillomavirus is a necessary cause of invasive cervical cancer worldwide," *The Journal of pathology*, vol. 189, no. 1, pp. 12–19, 1999.

[19] M. Dürst, L. Gissmann, H. Ikenberg, and H. Zur Hausen, "A papillomavirus dna from a cervical carcinoma and its prevalence in cancer biopsy samples from different geographic regions," *Proceedings of the National Academy of Sciences*, vol. 80, no. 12, pp. 3812–3815, 1983.

[20] P. G. Rose, B. N. Bundy, E. B. Watkins, J. T. Thigpen, G. Deppe, M. A. Maiman, D. L. Clarke-Pearson, and S. Insalaco, "Concurrent cisplatin-based radiotherapy and chemotherapy for locally advanced cervical cancer," *New England Journal of Medicine*, vol. 340, no. 15, pp. 1144–1153, 1999.

[21] L. J. Mango, "Computer-assisted cervical cancer screening using neural networks," *Cancer Letters*, vol. 77, no. 2-3, pp. 155–162, 1994.

[22] P. Bamford and B. Lovell, "Unsupervised cell nucleus segmentation with active contours," *Signal processing*, vol. 71, no. 2, pp. 203–213, 1998.

[23] C.-W. Chang, M.-Y. Lin, H.-J. Harn, Y.-C. Harn, C.-H. Chen, K.-H. Tsai, and C.-H. Hwang, "Automatic segmentation of abnormal cell nuclei from microscopic image analysis for cervical cancer screening," in *Nano/Molecular Medicine and Engineering (NANOMED), 2009 IEEE International Conference on*, pp. 77–80, IEEE, 2009.

[24] E. Kim and X. Huang, "A data driven approach to cervigram image analysis and classification," in *Color Medical Image analysis*, pp. 1–13, Springer, 2013.

[25] D. Song, E. Kim, X. Huang, J. Patruno, H. Muñoz-Avila, J. Heflin, L. R. Long, and S. K. Antani, "Multimodal entity coreference for cervical dysplasia diagnosis.," *IEEE Trans. Med. Imaging*, vol. 34, no. 1, pp. 229–245, 2015.

[26] L. Denny, L. Kuhn, A. Pollack, and T. C. Wright, "Direct visual inspection for cervical cancer screening," *Cancer*, vol. 94, no. 6, pp. 1699–1707, 2002.

[27] R. Sankaranarayanan, P. Basu, R. S. Wesley, C. Mahe, N. Keita, C. C. G. Mbalawa, R. Sharma, A. Dolo, S. S. Shastri, M. Nacoulma, *et al.*, "Accuracy of visual screening for cervical neoplasia: Results from an iarc multicentre study in india and africa," *International Journal of Cancer*, vol. 110, no. 6, pp. 907–913, 2004.

[28] T. Xu, H. Zhang, C. Xin, E. Kim, L. R. Long, Z. Xue, S. Antani, and X. Huang, "Multi-feature based benchmark for cervical dysplasia classification evaluation," *Pattern recognition*, vol. 63, pp. 468–475, 2017.

[29] J. N. Kapur, P. K. Sahoo, and A. K. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 273–285, 1985.

[30] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[31] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 773–785, 1979.

[32] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[33] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen, "Fuzzy c-means clustering with spatial information for image segmentation," *computerized medical imaging and graphics*, vol. 30, no. 1, pp. 9–15, 2006.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[35] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[36] D. H. Hubel and T. Wiesel, "Shape and arrangement of columns in cat's striate cortex," *The Journal of physiology*, vol. 165, no. 3, pp. 559–568, 1963.

[37] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.

[38] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[43] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[44] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[45] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[47] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[48] Kaggle, "Data," 2017. [Online; accessed 16-November-2017].

[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, Ieee, 2009.

[50] A. Karpathy, "Cs231n convolutional neural networks for visual recognition," *Neural networks*, vol. 1, 2016.

[51] R. Herrero, M. H. Schiffman, C. Bratti, A. Hildesheim, I. Balmaceda, M. E. Sherman, M. Greenberg, F. Cárdenas, V. Gómez, K. Helgesen, *et al.*, "Design and methods of a population-based natural history study of cervical neoplasia in a rural province of costa rica: the guanacaste project," *Revista Panamericana de Salud Pública*, vol. 1, pp. 362–375, 1997.

[52] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[54] R. Malviya, S. Karri, J. Chatterjee, M. Manjunatha, and A. K. Ray, "Computer assisted cervical cytological nucleus localization," in *TENCON 2012-2012 IEEE Region 10 Conference*, pp. 1–5, IEEE, 2012.

VITAE

- Name: Zaid Mohammed Atef Al-Yafeai
- Nationality: Yemeni
- Date of Birth: 28/12/1990
- Email: *alyafey22@gmail.com*
- Permenant Address: Sana'a, Yemen